

Sveučilište u Zagrebu  
Filozofski fakultet, I. Lučića 3  
Odsjek za informacijske znanosti

Kristina Vučković

# **MODEL PARSERA ZA HRVATSKI JEZIK**

*Doktorska disertacija*

Mentor: dr.sc. Zdravko Dovedan, red.prof.  
Komentor: dr.sc. Marko Tadić, izv.prof.

Zagreb, 2009.

*Zahvaljujem se svojim mentorima prof.dr.sc  
Zdravku Dovedanu i prof.dr.sc. Marku Tadiću.  
Veliko hvala i kolegama Željku Agiću, Vjeri Lopini i  
Boži Bekavcu na savjetima, pomoći i podršci.  
Posebno se želim zahvaliti i prof. Vjekoslavu Afriću  
koji se pojavio u pravom trenutku s pravim riječim.  
Zahvaljujem se i prof. Maxu Silberzteinu na  
pomoći oko izrade hrvatskoga modula za NooJ  
kao i na mnogobrojnim savjetima, strpljenju i  
bodrenju. Uz njega, ništa nije bilo nemoguće!*

*Ovaj rad posvećujem mami Fani, tetki Aidi i Marku...  
vi ste ono najbolje u meni...*

## SADRŽAJ

Sadržaj .....	2
1. UVOD .....	6
1.1. Obrazloženje teme .....	7
1.2. Razrada teme.....	11
2. TEORETSKA PODLOGA MODELA.....	12
2.1. Povijesni pregled.....	13
2.2. Izdvojene teorije gramatike prirodnoga jezika.....	18
2.2.1. Gramatike ovisnosti.....	18
2.2.2. Gramatike fraznih struktura .....	22
2.2.2.1. Lokalne gramatike.....	25
2.3. Pred-parsne procedure .....	30
2.3.1. Označivači vrsta riječi.....	30
2.3.1.1. Osnovna struktura i svojstva.....	31
2.3.1.2. Povijest označivača.....	34
2.3.1.3. Vrste označivača.....	35
2.3.2. Razdjelnici.....	37
2.4. Parseri.....	45
2.4.1. Povijest parsera .....	45
2.4.2. Definicije parsera.....	46
2.4.3. Prikaz parsera .....	48
2.4.4. Zadaci, ciljevi i karakteristike parsera .....	50
2.4.5. Struktura parsera .....	51
2.4.6. Metode parsanja.....	54
2.4.6.1. Parsanje upravljano gramatikom .....	56
2.4.6.2. Parsanje upravljano podacima.....	57
2.4.6.3. Kombinirano upravljano parsanje .....	61
2.4.7. Prikazi parsanja .....	63
2.4.7.1. Količina informacije u analizi .....	63
2.4.7.2. Potpunost analize.....	65
2.4.8. Algoritmi korišteni u parsanju.....	67

---

2.4.8.1. Earleyjev algoritam.....	67
2.4.8.2. CKY algoritam .....	68
2.4.8.3. EM algoritam.....	69
2.4.8.4. Ulazno-vanjski algoritam .....	70
2.4.8.5. Baum-Welch algoritam.....	70
2.4.8.6. Viterbijev algoritam .....	71
2.4.8.7. Algoritam lijevog kraja.....	71
2.4.8.8. LR - algoritam s lijeva.....	72
2.4.8.9. Generalizirani algoritam s lijeva .....	73
2.4.8.10. Algoritam s povratnim izvršenjem.....	74
2.4.8.11. Algoritam navigacijskog parsera .....	75
2.4.8.12. Algoritam kaskadnog parsera .....	76
2.5. Višeznačnost .....	78
2.6. Vrednovanje .....	84
2.6.1. Preciznost.....	85
2.6.2. Odziv.....	86
2.6.3. F-mjera .....	87
3. PRAKTIČNI DIO MODELA .....	88
3.1. Izvedba .....	89
3.1.1. Korpus za testiranje .....	90
3.1.1.1. Zlatni standard za razdjelnik.....	90
3.1.1.2. Zlatni standard za parser .....	91
3.2. Alati korišteni za izradu modela parsera za hrvatski jezik.....	93
3.2.1. NooJ.....	93
3.2.2. PrepareForNooj.....	99
3.2.3. XMLtoDIC.....	102
3.3. Leksička analiza .....	106
3.3.1. Leksikon.....	106
3.3.2. Flektivne gramatike .....	107
3.3.3. Opis svojstava .....	111
3.4. Sintaksna analiza .....	115
3.4.1. Imenska sintagma <NP> .....	118
3.4.2. Apozicijska sintagma <AP> .....	121
3.4.3. Atributska sintagma <AT> .....	123
3.4.4. Glagolska sintagma <VP> .....	125
3.4.5. Prijedložna sintagma <PP>.....	134

---

---

3.4.6. Koordinacija .....	137
3.4.7. Brojevi <M> .....	138
3.4.8. Rečenica <S> .....	139
3.4.9. Znakovi interpunkcije.....	140
3.4.10. Višeznačnosti .....	140
3.4.10.1. <NP> višeznačnosti.....	141
3.4.10.2. <AP> i <AT> višeznačnosti .....	148
3.4.10.3. <VP> višeznačnosti.....	149
3.4.10.4. <PP> višeznačnosti .....	151
3.4.10.5. Nepromjenjive vrste riječi .....	152
3.5. Rezultati.....	158
3.5.1. Nakon gramatike <i>VP_NP_PP.nog</i> : .....	159
1. rečenica .....	159
2. rečenica .....	160
3. rečenica .....	160
4. rečenica .....	161
5. rečenica .....	161
6. rečenica .....	162
7. rečenica .....	163
8. rečenica .....	163
9. rečenica .....	164
10. rečenica.....	164
3.5.2. Nakon gramatike <i>VP_DCobl_PCtyp.nog</i> : .....	165
8. rečenica .....	165
3.5.3. Nakon gramatike <i>VP_DC_Pcobl.nog</i> .....	166
2. rečenica .....	166
7. rečenica .....	167
3.5.4. Nakon gramatike <i>subjekt_predikat.nog</i> .....	167
2. rečenica .....	167
4. rečenica .....	168
9. rečenica .....	168
3.5.5. Nakon gramatike <i>recenica.nog</i> .....	168
3.5.6. Rezultati za prepoznavanje sintagmi.....	168
3.5.7. Rezultati za prepoznavanje padeža <NP> sintagmi.....	170
3.5.8. Kumulativni rezultati modela .....	171
4. ZAKLJUČAK i perspektive.....	173

---

---

5. RJEČNIK .....	175
6. LITERATURA .....	178
Prilozi .....	196
Prilog 1 – Dio zlatnog standarda za testiranje razdjelnika (uzorak2.not). .....	197
Prilog 2 – Dio zlatnog standarda za testiranje parsera.....	199
Prilog 3 – Opis imenskih paradigmi (hrCroNoun.nof) .....	201
Prilog 4 – Opis pridjevskih paradigmi (hrCroAdje.nof) .....	218
Prilog 5 – Opis glagolskih paradigmi (hrCroVerb.nof) .....	225
Prilog 6 – Opis paradigmi za brojeve i zamjenice (hrCroNumPro.nof).....	242
Prilog 7 – Primjeri parsnih stabala.....	248
Prilog 8 – Popis slika.....	253
Sažetak.....	255
SUMMARY .....	256
Životopis .....	257

## **1. UVOD**

### 1.1. Obrazloženje teme

U obradi prirodnoga jezika koristi se sofisticirana analiza u koju se integriraju morfologija, sintaksa, semantika, pragmatika i znanje o svijetu. Sve su te razine potrebne za izradu odgovarajuće aplikacije kojoj je osnovni cilj razumijevanje prirodnoga jezika. Takav program trebao bi reći ne samo pripada li unesena rečenica danom jeziku već i što ta rečenica znači. Kad bismo to uspjeli, mogli bismo govoriti o postojanju ‘umjetno inteligentnog’ programa za razumijevanje prirodnoga jezika. No, kako to Chabris<sup>1</sup> (Chabris, 1987:1) kaže, umjetna inteligencija je ipak znanost, a ne čarolija.

Prirodni jezici su po svojoj prirodi bezgranični što znači da mogu imati bezgranično mnogo rečenica. Ta činjenica govori u prilog tome da je sve te rečenice jednostavno nemoguće nabrojati. U potpunosti se slažem s Jensensonom, Heidornom (Jensens, Heidorn, 1989:93) i Karlsoonom (Karlsoon, 1995:10) kad kažu da je *pokušaj pisanja gramatike kojom bi se eksplicitno opisale sve i samo rečenice nekoga prirodnoga jezika praktično koliko i pronalaženje Svetoga Grala*. No, čovjek je ipak odlučio upustiti se u proces obrade prirodnoga jezika za što mu je potreban konačan popis specifikacija jezika, odnosno opis osnovnih struktura rečenica – članica danoga jezika. Tim se dijelom obrade bavi sintaksa prirodnoga jezika.

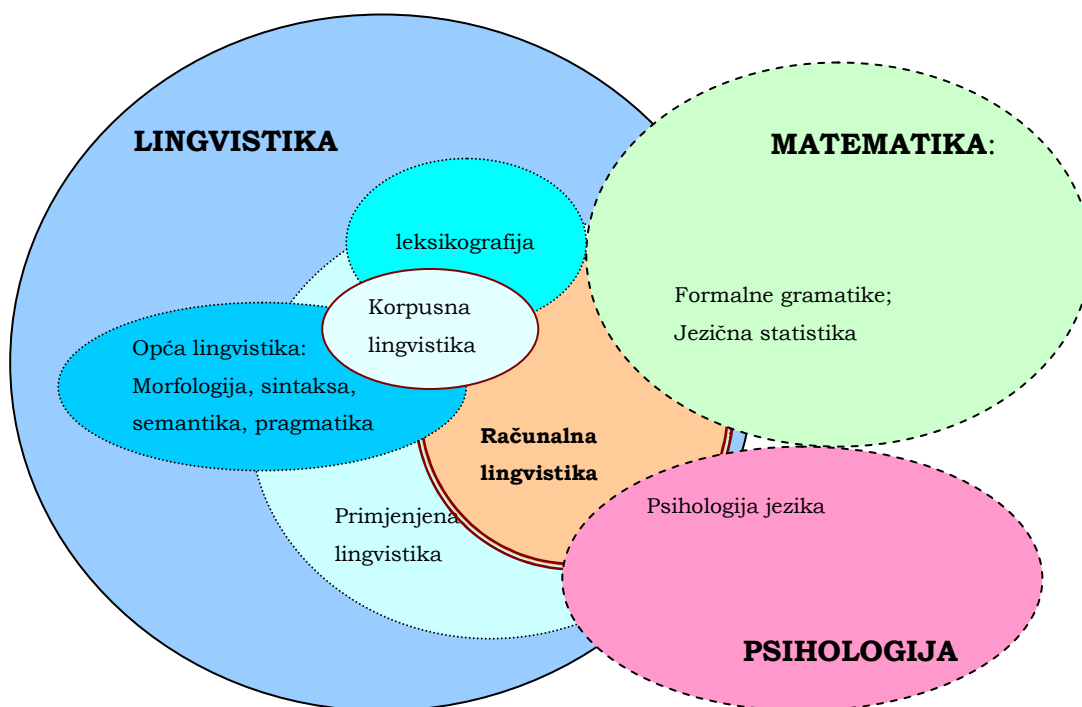
Obrada prirodnoga jezika našla je svoje mjesto u različitim granama pa je kao **obradu prirodnog jezika** nalazimo u računalnoj znanosti, kao **računalnu lingvistiku** u lingvistici, kao **prepoznavanje govora** u elektroinženjerstvu, te kao **računalnu psiholingvistiku** u psihologiji i kognitivnoj znanosti. Glavni zadaci obrade prirodnog jezika, kao računalne znanosti, su pripremiti sve potrebne izvore za obradu jezika, pronaći najbolje metode njihova povezivanja, izabrati alat za izradu programa te napisati potrebne algoritme. S druge strane, u centru računalne lingvistike je računalno obrađivanje prirodnoga jezika. Iz ovog opisa se može zaključiti kako

---

<sup>1</sup> C. Chabris je 2004. godine dobio Nobelovu nagradu iz psihologije.



računalna znanost predstavlja alat lingvistici u svrhu računalne obrade prirodnoga jezika. Za potrebe računalne lingvistike koriste se i drugi dijelovi lingvistike poput leksikografije, korpusne i opće lingvistike (morfologija jezika, sintaksa, semantika i pragmatika), ali i dijelovi drugih znanosti poput matematike (formalne gramatike i jezična statistika) i psihologije (psihologija jezika) (vidi Sliku 1).



Slika 1: Grafički prikaz područja računalne lingvistike

Iako se obradom prirodnoga jezika znanstvenici bave sad već više od 70 godina, vječna problematika koja se povlači od samih početaka uključuje sintaksu na jednom, a semantiku i pragmatiku<sup>2</sup> na drugom kraju, ne zaobilazeći pri tome ni referentnu višeznačnost, ulogu konteksta u obradi jezika i uklanjanje višeznačnosti riječi.

Sintaksa se analizira uz pomoć parsanja<sup>3</sup> ili sintaksne analize. Parsanje jezika je, u stvari, izvođenje odluka o procedurama jezika koje ga definiraju i prihvaćaju rečenice kao sastavnice, odnosno odbijaju rečenice

<sup>2</sup> Kad govorimo o sintaksi mislimo na opis struktura rečenica nekoga jezika, dok semantikom obilježavamo doslovno, a pragmatikom uporabno značenje tih rečenica.

<sup>3</sup> Parsanje dolazi od latinskog izraza '*pars orationis*' što znači 'vrste riječi'.

ne-sastavnice, toga jezika. Procesom parsanja na sintaksoj razini, rečenica se dijeli na svoje funkcionalne dijelove kojima se onda dodjeljuju rečenične službe. Ulaz u sintaktičku komponentu gramatike gotovi je oblik riječi 's pratećim kategorijalnim oznakama kojima se regulira njihovo uvrštavanje u rečeničnu strukturu' (Tadić, 1994:16).

Semantičkom<sup>4</sup> analizom pronalazi se opis značenja svake riječi pojedinačno koji se potom kombinira (na osnovi rečenične sintakse) tako da tvori značenje čitave rečenice. U pragmatici se u analizu uključuju i okolne rečenice kako bi se odredile referentne točke zamjenica i drugih višeznačnih sintaksoh objekata, te kako bi se potvrdila ispravnost semantičke analize ili pak utvrdilo neko dublje značenje rečenice. Također se uključuje i izvanjezični kontekst uporabe rečenica. No, ovim se aspektima jezične obrade u ovom radu neću detaljnije baviti.

U literaturi, kao i u praksi, pronalazimo različite kombinacije i redoslijede analiza u jeziku. Tako Lytinen (Lytinen, 1987:17) smatra da sintakso i semantičku analizu treba provoditi istovremeno, a ne da se rezultati sintaksoh parsanja prosljeđuju semantičkom interpretatoru. On, s toga, daje kratak opis sustava MOPTRANS<sup>5</sup> kao primjer sustava koji integrira obje analize.

Hoard (Hoard, 1998:206) pak vjeruje da bilo koji model za razumijevanje jezika treba imati 3 odvojene kategorije sposobnosti:

- kategorija za fonologiju, morfologiju i sintaksu;
- kategorija za semantiku;
- kategorija za pragmatiku i diskurs

te navodi čak 12 kriterija<sup>6</sup> koje svaka semantička teorija mora zadovoljavati kako bi se smatrala adekvatnom za razumijevanje jezika.

---

<sup>4</sup> Leksička semantika opisuje značenje riječi, a kompozicijska semantika opisuje kako se riječi kombiniraju da tvore šira značenja. (Jurafski, Martin, 2000:3)

<sup>5</sup> MOPTRANS je izveden od riječi MOP-based TRANSlator gdje kratica MOP predstavlja pakete organizirane memorije (engl. Memory Organization Packets). Sam MOPTRANS sustav koristi dva tipa znanja: apstraktno znanje (tj. konceptne veze) i paket-znanje (apstraktni nizovi događaja na koje se netko/nešto referira). Zadatak ovog sustava je bio da 'čita' novinske članke o terorizmu pisane na španjolskom i francuskom jeziku i prevede ih na engleski.

<sup>6</sup> *Partial intentionality; real-world validity; multi-valued logic; inferencig rules; semantic operators; coherence conditions; connectivity; generalized quantification; non-arbitrary relation to syntax; intentionality; higher-order constructs; amalgamation* (Hoard, 1998:209).

Govoreći o formalizmima temeljenima na ograničenjima (engl. *constraint based formalisms*), koji se koriste i u računalnoj i formalnoj lingvistici kao i u obradi prirodnoga jezika, Shieber (Shieber, 1992:6) upozorava na nespretnost opisivanja izraza na takav način da ga se smjesti u jednu od kategorija jer bi to zahtijevalo karakteriziranje svih svojstava distribucije tog izraza istovremeno.<sup>7</sup> *‘Ukratko, struktura informacije koja je dopuštena u formalizmu ne slaže se s prirodnom strukturom lingvističke informacije. Ovakvo razmišljanje dovodi nas do takvoga gramatičkoga opisa kojim se o jeziku razmišlja kao o odnosu između pojedinačnih nizova i informacija o njima, umjesto kao o funkciji između nizova i ispravnih prosudbi, ili kao klasificiranju nizova u konačan skup klasa’.* Zato svaki izraz u prirodnom jeziku ima prema Shieberu tri svojstva koja čine informaciju<sup>8</sup> o njemu potpunom:

- **modularnu strukturu** kojom se informacija o skupu promatra kao entitet s hijerhijskom strukturom, a ne kao neterminal;
- **parcijalnost** izraza koja omogućava ne postojanje svih informacija o svakom pojedinom izrazu na svakom stupnju indukcije u bezokolinskom formalizmu;
- **ravnotežu (ekvivalentnost)** izraza kojim se označava da informaciju o skupu određuje sustav uspoređujući i tražeći istovrijednosti pojedinačnih ograničenja elemenata skupa koja zasebno ne znače puno, no gledana u odnosu na cijeli skup ograničavaju identitet cijelog izraza.

Parser predstavlja sustav podataka i pravila kojima se želi što bliže simulirati ljudska sintaksna analiza jezika, u ovom slučaju hrvatskoga jezika. Cilj ovog rada je teorijski obraditi, definirati i implementirati model parsera za hrvatski jezik. Budući da takav model prije rada na parseru zahtijeva izradu razdjelnika a tek potom izradu parsera bilo je potrebno napraviti i razdjelnik za hrvatski jezik. Rezultati oba modela prikazani su u poglavljima koja slijede.

---

<sup>7</sup> No, što ako ponudimo samo mogućnosti pojavljivanja izraza, tj. odredimo njegove moguće okoline, a izbor radimo ovisno o danoj okolini?

<sup>8</sup> Shieber (Shieber, 1992:25) opisuje informaciju kao modularnu i hijerarhijski strukturiranu.

## **1.2. Razrada teme**

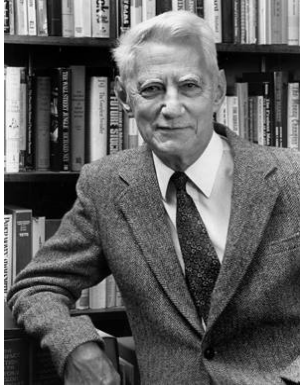
Rad je podijeljen na dvije izdvojene cjeline: teoretsku i praktičnu. Prva cjelina sadrži kratak povijesni pregled moderne obrade jezika. Nakon toga su definirane i izložene osnovne karakteristike dviju izdvojenih teorija gramatike prirodnoga jezika, a to su gramatika ovisnosti i gramatika fraznih struktura tj. lokalne gramatike. Slijedi kratak i sažet prikaz pred-parsnih procedura: označivača i razdjelnika. Poglavlje koje potom slijedi daje kratku povijest parsera, navodi različite definicije parsera koje nalazimo u literaturi, načine na koje možemo prikazati analizu parsera, zadatke i karakteristike svakog parsera kao i razloge zašto parsati. U tom se poglavlju nadalje navode i detaljnije opisuju strukture parsera, parsne metode, prikazi parsanja ovisno o količini informacija koju parser daje i o količini završenosti analize, te naposljetku i algoritmi koji se koriste u samom procesu sintaksne analize. Slijede još dva poglavlja prve cjeline a to su poglavlje koje obrađuje višeznačnosti u jeziku i konačno poglavlje koje daje pregled metoda vrednovanja koje sam koristila i za svoj predloženi model parsera.

Druga cjelina daje detaljan popis i opis gramatika kreiranih za sintaksnu analizu hrvatskih tekstova, dio modela za otklanjanje višeznačnosti, kao i analizu dobivenih rezultata. Na kraju rada dani su i prijedlozi za daljnji rad kojim bi se trebale otkloniti preostale višeznačnosti i podići efikasnost sustava. Rad završavam zaključkom kojim preciziram glavne točke rada.

## **2. TEORETSKA PODLOGA MODELA**

## 2.1. Povijesni pregled

Može se reći da je moderna obrada jezika svoj početak imala u



Claude E. Shannon

razdoblju nakon drugoga svjetskoga rata, dakle od 1940-ih do kraja 1950-ih godina. Priča se počela odvijati već 1943. godine kada su McCulloch i Pitts razvili jednostavan model neurona<sup>9</sup>. Nakon toga je 1948. godine Shannon<sup>10</sup> primijenio modele vjerojatnosti iz Markovljeva procesa na



Alan M. Turing

automate za jezik, da bi se 1950-ih

pojavili i prvi automati nastali na osnovi Turingova<sup>11</sup> modela



Stephen C. Kleene

iz 1936. na čemu i Kleene<sup>12</sup> 1951. godine

razvija automate konačnih stanja<sup>13</sup> i regularne izraze. Svaki se automat konačnih stanja može opisati uz pomoć regularnog izraza koji se definira kao formalni jezik za opis niza koji se traži.

Na osnovi Shannonova rada, Chomsky (Chomsky, 1956.) koristi konačna stanja za opis gramatike konačnih stanja koja može generirati jezik konačnih stanja. Svi navedeni modeli poslužili su stvaranju teorije formalnih jezika koja uključuje okolinske i bezokolinske gramatike koje se koriste za opis prirodnih jezika.

<sup>9</sup> Radi se o vrsti računalnog elementa kojega možemo opisati uz pomoć iskazne (propozicijske) logike.

<sup>10</sup> Claude Elwood Shannon (1916-2001) ili otac informacijske teorije je američki matematičar i inženjer elektrotehnike.

<sup>11</sup> Alan Mathison Turing (1912-1954) engleski matematičar, logičar i kriptograf koji se smatra ocem moderne računalne znanosti.

<sup>12</sup> Stephen Cole Kleene (1909-1994) je američki matematičar koji je pomogao u postavljanju teoretskih temelja računalne znanosti. Osnovao je granu matematičke logike, tzv. rekurzivnu teoriju koja je po njemu dobila i ime, Kleeneov rekurzivni teorem, isto kao i Kleeneova algebra, Kleeneova hijerarhija, Kleeneova zvijezda (pojavljivanje nekog znaka 0 ili više puta)...

<sup>13</sup> Automat konačnih stanja je mreža usmjerenih grafova koju čine stanja i označeni lukovi. Putanja automata je niz lukova koji vode od početnog do konačnog stanja. Razlikujemo jednostavne automate, kod kojih su lukovi označeni jednim simbolom **a**, od pretvarača (engl. *transducer*) kod kojih barem jedan luk ima par simbola **a:b**. Automat konačnih stanja ima tri osnovna svojstva: da je bez praznog simbola, da je deterministički (ni jedno stanje nema više od jednog luka s istom oznakom, da je minimalan (ne postoji niti jedan drugi automat koji opisuje iste putanje ali s manje stanja). Ukratko, možemo ga opisati kao alat za opis jezika.

Krajem 1950-ih dolazi do odvajanja obrade jezika i obrade govora u dvije zasebne paradigme: simboličku i stohastičku. Simbolička se bazirala na teoriji formalnih jezika, generativnoj sintaksi, algoritmima za parsanje na jednoj strani, te umjetnoj inteligenciji<sup>14</sup> na drugoj strani.

Stohastičkom paradigmom su se pretežno zaokupili statističari i inženjeri elektrotehnike kojima je bio interesantniji pristup optičkoga prepoznavanja slova uz korištenje velikoga rječnika i izračunavanja mogućega slijeda slova. Ova je metoda dobila i svoje ime pa je danas poznajemo kao Bayesovu<sup>15</sup> metodu. Stohastička paradigma je u godinama koje su slijedile, znatno doprinijela razvoju algoritama za prepoznavanje govora. Upravo iz ove paradigme krajem 1960-ih proizlazi i skriveni Markovljev model - HMM<sup>16</sup>.

Daljnji napredak donio je i nove paradigme. Jedna od njih je i logička paradigma na kojoj su nastali i Q-sustavi<sup>17</sup>, metamorfozna gramatika<sup>18</sup>, gramatike potpunih rečenica<sup>19</sup> te leksičko-funkcionalna gramatika (LFG).

Početak 1970-ih, stvara se i paradigma razumijevanja prirodnoga jezika. Njezin početak možemo zahvaliti Terryju Winogradu. Iako je prvi model sintakse prirodnih jezika Chomsky predložio još 1957. godine, prvi program za obradu prirodnoga jezika, poznat kao SHRDLU, napravio je upravo Winograd tek 1972. godine. Ovaj je program bio u stanju izvršiti naredbe primljene na prirodnome jeziku.



Terry Winograd

<sup>14</sup> Pojam 'umjetna inteligencija' prvi je put upotrijebio profesor matematike McCarthy i to ne tako davne 1956. godine.

<sup>15</sup> Bayesova metoda je ime dobila po britanskom matematičaru i fratr Thomasu Bayesu (1702-1761) koji je prvi formulirao Bayesov teorem.

<sup>16</sup> HMM je statistički model za rješavanje 3 tipa problema: a) izračunavanje vjerojatnosti određenog izlaznog niza na osnovi parametara modela; b) pronalaženje najvjerojatnijeg niza stanja koji bi doveli do određenog izlaznog niza na osnovi parametara modela; c) pronalaženje najvjerojatnijeg skupa prijelaza stanja na osnovi izlaznog niza.

<sup>17</sup> Ovu je metodu transformacija prema zadanim gramatičkim pravilima za potrebe obrade prirodnog jezika predložio Alain Colmerauer kasnih 1960-ih.

<sup>18</sup> Colmeraur, 1978.

<sup>19</sup> F. Pereira i D. Warren, 1980.

Prema D. Wilson i D. Sperberu (u njihovom radu iz 1986:158, *Relevance: Communication and Cognition*) zadatak razumijevanja prirodnoga jezika je konstruirati moguće interpretativne hipoteze o sadržaju ‘skupa pretpostavki’ i izabrati ispravnu hipotezu. Jedan od načina da se to napravi je nabrojati sve moguće hipoteze, usporediti ih i odabrati najbolju.



Deidre Wilson

Drugi pak način je da se prva hipoteza testira te jedino ako ne zadovoljava uvjete prelazi se u potragu za sljedećom hipotezom. U suprotnom se prihvaća prva koja je zadovoljila kriterije.

J. Allen (Allen, 1989:195) napominje

kako je odvajanje opisa lingvističke strukture od metoda za obradu jezika



Dan Sperber

rezultiralo približavanjem računalnih modela formalnim lingvističkim teorijama. On (Allen, 1989:198) smješta gramatičke formalizme u klasu unifikacijskih gramatika<sup>20</sup>. Zajednička osnova tih formalizama su bezokolinske gramatike.

Adriaens i Hahn (Hahn, Adriaens, 1994.) promatraju obradu prirodnoga jezika kroz više perspektiva: paralelnu, distribuiranu, interaktivnu (misli se na dvosmjerni protok informacija) i vezanu perspektivu.

Do danas je predloženo i više modela objektno-orijentiranoga pristupa obrade prirodnoga jezika. Tako uz Philipsa i Hendlera (Philips i Hendler, 1982.) koji koriste pojam objekta za organizaciju sustava za obradu prirodnoga jezika i smatraju ga pogodnim za sintaksnu i semantičku obradu, nalazimo još i objektno-orijentirane sustave poput WEP-a, ParseTalka i WCFA-a.

WEP (engl. *Word Expert Parsing*) predlažu Small i Rieger (Small i Rieger, 1982.) kao sustav za ekspertno procesiranje riječi gdje je svaka riječ

<sup>20</sup> Unifikacijske gramatike unificiraju iste vrijednosti atributa elemenata analize po principu presjeka skupova tj. ako dva elementa imaju sve iste attribute i njihove vrijednosti, oni se svi preuzimaju (the fish [sing, plu] ∩ [sing, plu] = [sing, plu]), ako imaju zajedničke neke vrijednosti atributa, samo se ti preuzimaju (the cats [sing, plu] ∩ [plu] = [plu]), i ako nemaju zajedničku niti jednu vrijednost atributa unifikacija nije uspjela (a cats [sing] ∩ [plu] = ∅).



objekt. Ova je teza motivirana mišljenjem da je ljudsko znanje o jeziku organizirano prvenstveno kao znanje o riječima, a ne znanje o pravilima. Nedostaci ovog pristupa su nemogućnost sustavnoga dodavanja gramatičkoga znanja kao i nemogućnost provođenja induktivnog zaključivanja nad skupovima leksičkih podataka.

ParseTalk je objektno-orijentiran sustav za simultano parsanje koji koristi posebnu paradigmu za kontrolu analize uz pomoć *word actorsa* koji međusobno komuniciraju, tj. razmjenjuju informacije koje pokreću određene događaje (Hahn, Schacht, Broker, 1994). *Word actorsi*, koji su međusobno vezani nekim relacijama ovisnosti, povezuju se u specijalizirane *phrase actorse* koji sadrže sve informacije o toj sintagmi. Gramatika koju ovaj sustav koristi temelji se na binarnim relacijama između riječi, odnosno riječi-glave i riječi koja ju opisuje, modificira ili dopunjuje. Svojstva riječi opisana su preko klasa riječi kojima pripadaju, a uključuju posebna morfosintaksna svojstva, valenciju i točan raspored pojavljivanja. Može se smatrati centralnim dijelom jezične obrade kada se koristi u sustavima koji imaju mogućnost stalnog dodavanja znanja o novim konceptima za vrijeme provođenja analize teksta.

WCFA (engl. *Word Class Functions Analysis*) je paradigma za sintaksno-semantičku analizu prirodnog jezika (Helbih, Hartrumpf, 1997). Prvenstveno je orijentirana na riječ, a centralno mjesto zauzimaju funkcije klasa riječi. Svaka riječ ima dvije faze aktivnosti: otvaranje i zatvaranje očekivanoga slijeda. Na taj način svaka riječ stvara određena očekivanja onoga što može doći poslije nje. Nakon faze zatvaranja očekivanoga, 'podešava' se sintaksna i semantička informacija kojom ta riječ doprinosi cijelom nizu.

Krajem 20. stoljeća sve je više modela temeljenih na vjerojatnosti, odnosno upravljanih podacima, koji se koriste ne samo za prepoznavanje govora i pronalaženje obavijesti već i za POS označivanje i parsanje.

U odnosu na prethodno stoljeće, danas su strojevi brži i pouzdaniji, a njihova memorija raste iz dana u dan što uvelike pomaže polje računalne obrade prirodnog jezika. Istovremeno, sve veća proširenost Interneta, odnosno povećanje sadržaja na njemu, osigurava nove izvore tekstova

pogodnih za računalnu obradu. Tako nastaju i prve velike tekstne zbirke s različitim sintaksnim, semantičkim i pragmatičkim obilježjima.

## 2.2. Izdvojene teorije gramatike prirodnoga jezika

Iako je do sada predložen znatan broj različitih gramatika za obradu prirodnoga jezika, ovdje ću izdvojiti i detaljnije prikazati dva gramatička formalizma: gramatiku ovisnosti i lokalne gramatike.

### 2.2.1. Gramatike ovisnosti

Teoriju gramatike ovisnosti predložio je u svojoj teoriji strukturalne sintakse iz 1959. godine Lucien Tesnière<sup>21</sup>, ali se njezini korijeni mogu vidjeti i u Pāṇinijevoj sanskrtskoj gramatici<sup>22</sup>, arapskoj gramatici osmog stoljeća te njemačkoj gramatici devetnaestog stoljeća (Mel'čuk, 2003). Različiti su autori koristili pojmove gramatike ovisnosti u svojim teorijama pa ih nalazimo u Fillmoreovoj padežnoj gramatici iz 1968., Hudsonovoj *Word Grammar* iz 1984., Sgallovom funkcionalnom generativnom opisu i Hellwigovoj gramatici unifikacijske ovisnosti iz 1986., Starostinom *Lexicaseu* iz 1988. i drugima<sup>23</sup>.

**Gramatika ovisnosti** je sintaksna teorija koja strukturu rečenice definira ovisnostima koje postoje među riječima tj. između riječi – glave<sup>24</sup> i njezinih ovisnica<sup>25</sup>. Mel'čuk (Mel'čuk, 2003:10) opisuje ovisnost kao nesimetričnu relaciju istog tipa kao što je i logička implikacija. To znači da jedan element 'očekuje' postojanje nekog drugog elementa u određenom obliku, ali obrnuto ne vrijedi. Prema tome zaključujemo da se radi o usmjerenom odnosu glava<sup>26</sup> -> ovisnica.

<sup>21</sup> L. Tesnière je francuski lingvist (1893.-1954.) koji se bavio slavenskim jezicima i sintaksnom teorijom kojom bi ih opisao.

<sup>22</sup> Vidi više u (Vučković, 2004).

<sup>23</sup> Vidi više u (McDonald, Nivre (2007), Vučković (2004) Mel'čuk(2003)).

<sup>24</sup> U literaturi se još uz pojam 'head' koriste pojmovi 'governor', 'ruler' i 'regent'.

<sup>25</sup> U literaturi se još uz pojam 'dependent' koristi pojam 'modifier' i 'satellite'.

<sup>26</sup> Mel'čuk umjesto pojma 'glava' preferira pojam 'upravitelj' (engl. *governor*) jer smatra da pojam 'glava', koji je preuzet iz fraznih struktura, nosi sa sobom negativnu konotaciju kojom se implicira sastavnica.

Tesnière navodi šest aksioma koji definiraju strukturu usmjerenoga grafa, odnosno stabla, a koje ovdje preuzimam iz (Järvinen, Tapanainen, 1998:6):

- osnovni element sintaksnog opisa je nukleus;
- sintaksna se struktura sastoji od veza između nukleusa;
- veza je binarna funkcijska relacija između nadređenog i podređenog pojma;
- svaki nukleus je čvor u sintaksnom stablu i ima samo jedan nadređeni pojam;
- nadređeni pojam koji ima nula ili više ovisnika predstavlja cijelu granu;
- najviši nadređeni pojam je središnji čvor rečenice.

Mjesto u rečenici i gramatički oblik riječi ovisi o poziciji i obliku druge riječi u toj rečenici. Osnovna ideja ovisnosti temelji se na mišljenju da se rečenične sintaksne strukture sastoje od binarnih asimetričnih relacija među riječima dane rečenice. Glavni zadatak ove teorije je uspostavljanje ovisnih relacija i određivanje koje su riječi glava, a koje ovisnice glave unutar svake detektirane relacije. Neki autori dopuštaju postojanje samo jedne glave (npr. Tesnière i Mel'čuk) dok drugi pak dopuštaju postojanje više glava (npr. Hudson). Za opis sintagmi u hrvatskom jeziku, ja sam se odlučila za postojanje samo jedne glave pa se na tom pravilu temelje i gramatike za prepoznavanje imenske, prijedložne i glagolske sintagme. Tako će jedna imenica biti glava imenske sintagme, prijedlog će biti glava prijedložne sintagme a glavni glagol će biti glava glagolske sintagme.

Prilikom određivanja glave i ovisnice unutar neke konstrukcije primjenjuju se sljedeći sintaksno/semantički kriteriji koje ovdje prenosim iz (McDonald, Nivre, 2007.):

- glava određuje sintaksnu kategoriju i obično je može zamijeniti,
- glava određuje semantičku kategoriju dok ovisnica daje semantičku specifikaciju,
- glava je obavezna dok je ovisnica izborna,

- glava bira ovisnicu i određuje njezinu izbornost (tj. određuje je li ovisnica izborna ili obavezna),
- oblik ovisnice ovisi o glavi, (tj. ako je imenica u jednini ž.r. i padež koji je opisuje će preuzeti taj oblik),
- glava određuje linearnu poziciju ovisnice.

Tri su osnovna tipa ovisnosti:

- **bilateralna ovisnost** – pojavljivanje jednog elementa ovisi o pojavljivanju drugog elementa tj. niti se glava može pojaviti bez ovisnice niti ovisnica bez glave (npr. prijedlog + objekt);

✍ *On je išao **u šumu**.*

✍ *\*On je išao **u**.*

✍ *\*On je išao **šumu**.*

- **unilateralna ovisnost** – glava se može pojaviti bez ovisnice, ali ne i obrnuto;

✍ ***Veoma smiješna priča** je ispričana.*

✍ *\* **Veoma priča** je ispričana.*

✍ ***Smiješna priča** je ispričana.*

✍ *\***Smiješna** je ispričana..*

✍ ***Priča** je ispričana.*

- **koordinirana ovisnost** – dva ili više vezanih elemenata zadržavaju isti status glave;

✍ ***Dječak i djevojčica** brali su jagode.*

✍ ***Sretni i veseli** vratili su se kući.*

✍ *Prije spavanja su se **okupali i večerali**.*

Neki autori poput Mel'čuka razlikuju dva osnovna tipa ovisnosnih relacija odnosno odnosa: sintagmatski odnosi (morfološka, sintaksna i semantička ovisnost) i paradigmatski odnosi (sinonimija, antonimija)<sup>27</sup>:

- **morfološka ovisnost** – kada morfološki oblik jedne riječi ovisi o morfološkom obliku druge riječi (npr. imenice u genitivu traži i pridjev koji je opisuje da bude u genitivu, pa kažemo da oblik pridjeva ovisi o obliku imenice);
- **sintaksna ovisnost**<sup>28</sup> – kada između dvije riječi postoji:
  - sintaksna povezanost (kako bi se utvrdilo postojanje ovisnosti između dviju riječi),
  - sintaksna dominancija (kako bi se utvrdila orijentacija ovisnosti između dviju riječi) i
  - sintaksni tip ovisnosti;
- **semantička ovisnost** – kada je značenje jedne riječi argument značenju druge riječi (npr. argument predikata semantički ovisi o predikatu jer mu predikat svojim značenjem otvara mjesto).

Kad govorimo o ovisnosti, u stvari govorimo o gramatičkim vezama između dviju ili više rečeničnih jedinica. Tako ovisnost može biti ona između glagola i imenskoga i/ili prijedloznoga skupa. Posebnu kategoriju izraza imaju funkcijske riječi poput članova (*the*, *a*, *an*) u engleskom jeziku, pomoćnih glagola, te strukture koje sadržavaju prijedložne izraze za koje ne postoji točno utvrđen princip njihova analiziranja<sup>29</sup>.

Kod mjesnih ovisnosti, ovisne jedinice ne mogu biti previše udaljene jedna od druge. Za ovaj tip ovisnosti karakteristična su tri različita odnosa:

- **potkategorizacija** – prisutnost jedne jedinice zahtijeva prisutnost neke druge jedinice (npr. glagol koji zahtijeva objekt);

---

<sup>27</sup> Vidi više u (Mel'čuk, 2003:9).

<sup>28</sup> Mel'čuk je smatra najvažnijim tipom ovisnosti.

<sup>29</sup> Npr. neke gramatike ovisnosti prijedlog tretiraju kao glavu imenici, a druge pak smatraju da je imenica glava čija je ovisnica prijedlog.

- **upravljanje** – prisutnost jedne jedinice zahtijeva poseban oblik druge jedinice (npr. prijedlog 'u' zahtijeva ime u lokativu);
- **sročnost** – oblik jedne jedinice zahtijeva poseban oblik druge jedinice (npr. slaganje subjekta i predikata u rodu, broju i licu).

Postavlja se naravno pitanje, je li samo određivanje ovisnosti među riječima dovoljno za analizu sintaksne strukture prirodnoga jezika. Nivre (Nivre 2005:24) smatra ovisne strukture dovoljno ekspresivnima za korištenje pri obradi prirodnog jezika, ali i dovoljno ograničavajućima za provedbu potpunog parsanja s visokom točnošću i efikasnošću. Mel'čuk smatra kritike za ovisnosni pristup sintaksi neopravdanima vjerujući da su one nastale zbog nesnalaženja među različitim tipovima ovisnosti te ih opovrgava primjerima i pojašnjenjima navedenima u (Mel'čuk, 2003).

Neki autori poput (Nivre, 2005:24) vjeruju da je parsanje temeljeno na gramatici ovisnosti dobar izbor procesa parsanja za jezike s varijabilnim poretком riječi kao što je i hrvatski. Gramatika ovisnosti korištena je i za jezike poput japanskog, kineskog i korejskog (Tom Lai, Huang, 1994.).

Dva pristupa sintaksnom parsanju koja se zasnivaju na gramatici ovisnosti su:

- pristup upravljan gramatikom (engl. *grammar-driven approach*);
- pristup upravljan podacima (engl. *data-driven approach*).

Njih ću detaljnije opisati u poglavlju 2.4.6.

### 2.2.2. Gramatike fraznih struktura

Možda i nije tako iznenađujuće da ideja o temeljenju gramatike na konstitucijskim strukturama dolazi iz područja psiholingvistike. Naime, to je još 1900. predložio Wilhelm Wundt, ali je trebalo skoro više od pola stoljeća da Chomsky (1956.) a kasnije i Backus (1959.) i Naur (1960.) formaliziraju tu ideju (Jurafsky, Martin, 2000:387). Na osnovu njihovih postavki, radovi u

obradi prirodnoga jezika koji su uslijedili temeljili su se većinom na bezokolinskim gramatikama tj. na gramatikama fraznih struktura.

Najraniji jezični modeli opisivali su gramatičke kategorije, odnosno formalizirali su rečenične forme, mijenjanjem riječi s njihovim gramatičkim kategorijama.

✍ Marko piše. => Imenica Glagol.

Modeli fraznih struktura<sup>30</sup> predstavljaju direktnu refleksiju gramatičke analize. Te je modele Chomsky 1957. godine smjestio u bezokolinske gramatike. Ovisno o obliku svojih produkcija, razlikujemo 4 tipa gramatika Chomskog koje tvore hijerarhiju Chomskog:

- regularna gramatika (gramatika tipa 3);
- bezokolinska ili gramatika fraznih struktura (gramatika tipa 2);
- okolinska gramatika (gramatika tipa 1);
- gramatika bez ograničenja (gramatika tipa 0).

S obzirom na hijerarhijski poredak ovih gramatika, zaključujemo da je svaka gramatika tipa 3 istovremeno i tipa 2, odnosno, svaka gramatika tipa 2 istovremeno je i tipa 1, te je svaka gramatika tipa 1 istovremeno i tipa 0. Jednostavnost jezika kojeg ove gramatike koriste, opada od gramatike tipa 3 do gramatike tipa 0 pa je jezik za gramatiku bez ograničenja najsloženiji. Formalni opisi ovih gramatika dani su u (Jurafski, Martin, 2000.), (Dovedan, 2003.) i (Martin-Vide, 2003.).

Gramatike fraznih struktura omogućavaju spajanje niza riječi u jednu skupinu, odnosno, sintagmu koristeći formalizme u obliku bezokolinskih pravila tj. produkcija. Pravila (Slika 2) se mogu prikazati i uz pomoć stabla izvođenja, odnosno, parsnog stabla (Slika 3). Pravila pokazuju načine na koje je određene kategorije riječi u jeziku moguće međusobno spajati kako bi dobili valjanu rečenicu u zadanom jeziku. Za pisanje pravila koristimo skup

<sup>30</sup> Sama ideja kojom se gramatika temelji na strukturi sastavnica datira još iz 1900. a predložio ju je psiholingvist Wilhelm Wundt. No, tek je 1956. Chomski ovu ideju formalizirao. (Jurafski, Martin, 2000:387).



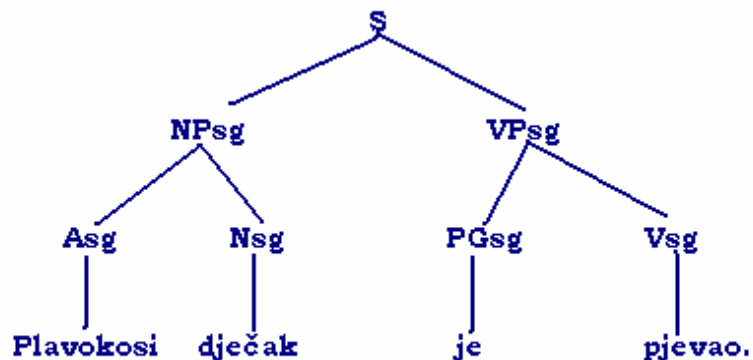
terminala i neterminala na način da se neterminali mogu naći i s lijeve i s desne strane pravila dok se terminali mogu naći samo s desne strane pravila. Terminali su konkretne riječi nekog jezika dok su neterminali simboli kojima opisujemo jedan terminal ili mogućnosti spajanja više terminala. Tako je cijeli flektivni leksikon promijenjivih vrsta riječi i cijeli leksikon nepromijenjivih vrsta riječi skup terminala za opis hrvatskoga jezika. Skup neterminala nalazi se u *properties.def* datoteci (vidi poglavlje 3.3.3.).

```

S -> NPsg VPsg
NPsg -> Nsg
NPsg -> Asg Nsg
VPsg -> Vsg
VPsg -> PGsg Vsg
Nsg -> dječak
Asg -> plavokosi
Vsg -> pjevao

```

Slika 2: Prikaz gramatike uz pomoć pravila



Slika 3: Prikaz uz pomoć stabla izvođenja

Kao što je bolje vidljivo iz prikaza stablom izvođenja, kada se govori o gramatici fraznih struktura govori se o hijerarhijskom prikazu u kojem se hijerarhijski više kategorije (npr. S, NP, VP) sastoje od jednog ili više manjih sintaktičkih dijelova. Oni se nadalje mogu sastojati od jednog ili više još manjih sintaktičkih dijelova. Bitno je pri tome da se svaka sintaktički viša struktura sastoji jedino od sintaktički dozvoljenih dijelova.

Mogućnost dodavanja dodatnih gramatičkih opisa (rod, broj, padež, vrijeme), odnosno pisanje pravila kojima se postiže slaganje riječi unutar sintagme (npr. pridjev i imenica u nominativu jednine) i među sintagmama (npr. NP i VP u muškom rodu jednine), čini gramatike fraznih struktura primjenjivima i na opis prirodnih jezika. Ovo naravno vrijedi samo ako se radi o konfigurativnim jezicima poput hrvatskoga u kojima se riječi u rečenici spajaju prema strogim pravilima gramatike.

#### **2.2.2.1. Lokalne gramatike**

Modeli fraznih struktura, uključujući i transformacijske gramatike koje je predložio Z. S. Harris, ipak nisu uspjele opisati sve iznimke prirodnog jezika. Upravo kao model za rješenje iznimki, Gross 1989. godine predstavlja lokalne gramatike i definira ih kao gramatike konačnih stanja koje predstavljaju skupove izraza prirodnoga jezika.

Gross (Gross, 1995a) je smatrao da je neke izraze, bilo iz sintaksnih ili semantičkih razloga, bolje opisati kao sastavne obiteljske nizove (engl. *families of strings*), a ne kao zasebne ulaze u leksikonu. Obiteljski nizovi mogu biti povezani preko barem jednog svog značenja ili pak mogu biti sinonimi, no svi članovi jednog obiteljskog niza moraju imati zajednička formalna (leksička i sintakсна) svojstva. Upravo je te obiteljske nizove Gross nazvao lokalnim gramatikama vjerujući da je u prirodi jezika spajanje različitih lokalnih modela konačnih stanja.

Kod jezika s polu-slobodnim (npr. njemački, korejski) i slobodnim poretком riječi (npr. hrvatski, češki) poredak određenih sintagmi, odnosno obiteljskih nizova, nije tako strogo određen kao što je to, na primjer, slučaj u engleskom jeziku. Sposobnost rečeničnih elemenata da se pojavljuju na različitim mjestima u rečenici naziva se premetanje (engl. '*scrambling*'). Iako bi se moglo zaključiti da su samo flektivni jezici predodređeni za premetanje istraživanja pokazuju da to i nije uvijek slučaj. Tako npr. nizozemski koji nema padeža dozvoljava premetanje dok ga islandski ne dozvoljava iako se

radi o jeziku s padežima. U literaturi (Becker et al., 1991. i Kang, 2005.) nalazimo dva tipa premetanja:

- među rečenično premetanje (engl. *Long-distance scrambling*) (Becker et.al., 1991. i Kang, 2005.) ili ne-lokalno premetanje (engl. *non-local scrambling*) (Hockenmaier, Young, 2008.)) - neki je argument glagola iz zavisne rečenice prebačen u glavnu rečenicu;
- unutar rečenično premetanje (engl. *Clause-internal scrambling*) ili lokalno premetanje (engl. *Local scrambling*) - dozvoljen je bilo koji poredak glagolskih argumenata, kao npr. u rečenicama:

- ✍ *Sinoć u 8 sati na mostu sam susrela dvije stare prijateljice.*
- ✍ *Dvije stare prijateljice susrela sam na mostu sinoć u 8 sati.*
- ✍ *Sinoć u 8 sati susrela sam dvije stare prijateljice na mostu.*
- ✍ *Na mostu sam susrela dvije stare prijateljice sinoć u 8 sati.*
- ✍ *Susrela sam dvije stare prijateljice sinoć u 8 sati na mostu.*

Lokalne gramatike ili lokalna pravila mogu opisati slijed riječi unutar niza, odnosno, slijed riječi za određene podskupove riječi u rečenici koji su neovisni o ostatku rečenice. Mason (Mason, 2004.) ih opisuje kao “*princip opisa sintaksnog ponašanja grupe individualnih elemenata koji su u nekom odnosu, ali se njihove sličnosti ne mogu jednostavno prikazati uz pomoć pravila fraznih struktura*”.

Lokalnom se gramatikom može opisati i niz koji se nikako ne smije pojaviti u određenoj kombinaciji (što se vidi iz primjera u Mohri (1994.) i Tapanainen, Järvinen (1997.)). I jedan i drugi pristup mogu se prikazati automatom konačnih stanja. Više manjih lokalnih gramatika može se spajati uz pomoć unije.

Lokalne gramatike mogu se primijeniti i kod zaleđenih kombinacija, odnosno kod opisa idioma, kolokacija, metafora, prenesenih značenja itd.

Konačan automat koji opisuje zaleđene kombinacije zahtijeva leksikon u kojem su imenice klasificirane po semantičkim svojstvima (npr. jezik, živo biće, čovjek, grupa...).

✎ *(broadly + generally + roughly) speaking*

✎ *(stupidly + surprisingly) enough*

✎ *(English) speaking (student)*

Tako, na primjer, kao što se vidi iz našeg posljednjeg primjera, na mjestu riječi '*English*' može doći bilo koji naziv jezika (French, German, Croatian, ...) odnosno imenica koja ima svojstvo +Jezik. Na mjestu riječi '*student*' može doći bilo koja imenica čije je svojstvo +Human ili +HumanGroup (npr. child, woman, Parliament). Riječ '*speaking*' u ovoj je kombinaciji obavezna i nije potrebno da ima neko +Svojstvo jer se navodi u samom pravilu za ovaj zaleđeni izraz.

✎ *sasvim (slučajno, iznenada, nenadano)*

✎ *(jučer, danas, sutra, petak) (ujutro, u podne, navečer, predvečer)*

✎ *(englesko, francusko, kajkavsko) govorno područje*

U prvom primjeru za hrvatski jezik imamo prilog '*sasvim*' kao nepromjenjivi dio zaleđenog izraza iza kojeg može doći bilo koji drugi prilog s oznakom +Način. U drugom primjeru za hrvatski jezik oba dijela izraza su promjenjiva pa imamo bilo koju imenicu s oznakom +Dan iza koje slijedi bilo koji prilog s oznakom +DobaDana. U posljednjem primjeru za hrvatski jezik imamo obrnutu situaciju od prvog primjera pa je samo prvi dio promjenjiv i

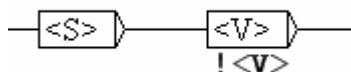
na tom mjestu može doći bilo koja imenica s oznakom +Jezik ili +Narječje (hrvatsko, grčko, kajkavsko) dok je izraz ‘govorno područje’ obavezan.

Interpretacija zaleđenih izraza gotovo je uvijek ispravnija od interpretacije pojedinačnih dijelova tog izraza (Roche, 1997:267) pa joj se u analizi treba dati prednost pred ostalim interpretacijama. Gross napominje kako je broj zaleđenih izraza u prirodnom jeziku dosta velik pa predlaže generalizaciju tih izraza (Gross, 1993.) vjerujući da će se na taj način bitno poboljšati domet parsera.

Maria Todorova je u (Todorova, 2006.) formalizirala idiomatske strukture za bugarski jezik. Ona idiomatsku strukturu definira kao lingvističku jedinicu koju čine dvije ili više riječi bez kompozicijskog značenja. Dijeli ih na zaleđene i polu-zaleđene izraze, od kojih prva grupa ima definiran oblik i poredak dok je u drugoj grupi barem jedan, a moguće i više elemenata, djelomično ili potpuno promjenjivo.

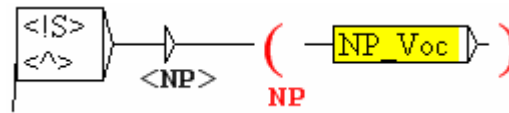
Lokalne gramatike će se u ovom radu koristiti za opis imenskih, apozicijskih, atributskih, glagolskih i prijedložnih sintagmi, koordinaciju i određivanje argumenata glavnog glagola. Većina gramatika koje sam opisala opisuju nizove koji se moraju pojaviti u određenoj kombinaciji. Izuzetci su 3 gramatike koje koriste negaciju na sljedeće načine:

- **negacija u izlazu** – izlazni niz ima u sebi negaciju. Tako se u našem primjeru definira da se glagol (V) ne smije pojaviti iza prijedloga (S). Odnosno, ako se iza prijedloga (S) nađe riječ koja se zasebno može interpretirati kao glagol (V), tada se ovom gramatikom ta interpretacija uklanja (vidi Sliku 4);



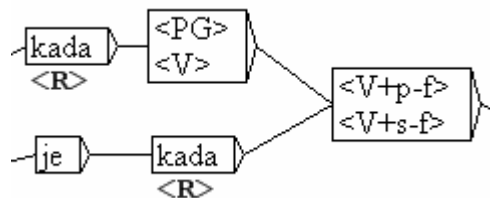
Slika 4: Negacija u izlazu

- **negacija u ulazu** – ulazni niz ima u sebi negaciju:
  - **negacija cijele klase** - određuje se da je imenska sintagma (NP) u vokativu samo ako ispred nje nije prijedlog (S) (vidi Sliku 5);



Slika 5: Negacija cijele klase

- o **negacija dijela klase** – određuje se da riječ 'kada', koja može biti interpretirana kao imenice (N) i kao prilog (R), dobiva interpretaciju priloga u slučaju da je odmah iza nje pomoćni glagol a odmah potom i glavni glagol u množini koji ne smije biti u ženskom rodu ili ako se nađe između pomoćnog glagola i glavnog glagola koji ponovo može biti i u množini i u jednini ali ne i u ženskom rodu (vidi Sliku 6);



Slika 6: Negacija dijela klase

## 2.3. Pred-parsne procedure

Zadatci iz obrade prirodnoga jezika koji prethode parsanju razvijali su se u nekoliko pravaca: određivanje rečenične granice (engl. *sentence boundary disambiguation* = SBD), lematizacija, označivanje rečeničnih dijelova ili POS označivanje (engl. *part-of-speech tagging* = POS tagging), razdjeljivanje rečenice na sintagme pomoću razdjelnika. Tek nakon njih slijedi parsanje, a onda i razumijevanje teksta. Ovdje ćemo ukratko opisati POS označivanje i razdjeljivanje rečenice na označene sintagme.<sup>31</sup>

### 2.3.1. Označivači vrsta riječi

Označivanje rečeničnih dijelova smatra se najnižom standardnom procedurom koja prethodi svim ostalim analizama. Cilj je ove procedure da svakoj riječi u rečenici pridruži najprikladniju morfosintaksnu kategoriju što znači da jedna riječ u danoj rečenici može imati samo jednu oznaku<sup>32</sup> (npr. imenica, zamjenica, glagol ili prijelazni glagol, glagol u 3.l.j., glagol u prezentu, imenica u genitivu, itd.). Ako se pak dogodi da je jednoj riječi moguće pripisati više oznaka, radi se o riječi koja može imati više značenja, ali i više funkcija u rečenici. Takve riječi u hrvatskome jeziku (ali i u većini drugih jezika) najčešće imaju i različit izgovor, no zbog nekorištenja oznaka za naglasak u pisanju hrvatskih tekstova, tu razliku ne možemo iskoristiti za ovakvu analizu.

Potpuno označivanje podrazumijeva rješavanje morfološke višeznačnosti, odnosno, predstavlja proces **odabira** 'ispravne' morfološke

<sup>31</sup> Određivanjem rečenične granice se ovdje neću dalje baviti jer je ono detaljno opisano u (Boras, 1998).

<sup>32</sup> Penn treebank koristi 20 funkcionalnih oznaka: *non-specific adverbial, benefactive, it-cleft, closely related, direction, dative, extent, headline, logical subject, location, manner, nominal, predicate, purpose, locative complement of 'put', subject, temporal, topic, title, vocative* (Blaheta, Charniak) i 45 POS oznaka (pridjev, komparativ pridjeva, superlativ pridjeva, prilog, komparativ priloga, superlativ priloga, čestica, modalni glagol, strana riječ, jednina imenice, množina imenice, jednina vlastite imenice, množina vlastite imenice, itd.), dok Brown korpus ima čak 87 POS oznaka (Jurafsky, Martin).

interpretacije za svaku pojavnicu u danom kontekstu (Oflazer, 2006:89). S druge strane, Voutilainen POS označivanje opisuje kao program kojim se svakoj riječi u tekstu **pridružuje** kontekstno prikladan gramatički opis. Kao ulazni niz, označivač uzima pojavnicu i automatski joj dodjeljuje opisne oznake koje sad, zajedno s ulaznim nizom, čine izlazni niz.

Najjednostavniji POS označivači su digramski i trigramski modeli koji zahtijevaju velike označene tekstove za vježbu kao što to zahtijevaju i transformacijski označivači.

Razvojem i usavršavanjem parsera, Charniak (Charniak, 1997.) vjeruje da slabi uloga označivača kao predprocesora parsanju. To potkrepljuje činjenicom da parseri imaju bolji učinak ako sami rade označivanje na neoznačenom tekstu. No, označivači svoje mjesto mogu naći u nekim drugim aplikacijama kao što su odgovaranje na upite, crpljenje obavijesti, sinteza govora, prepoznavanje govora, strojno prevođenje, pretraživanje velikih tekstnih baza itd.

#### 2.3.1.1. Osnovna struktura i svojstva

U literaturi se POS<sup>33</sup> označivanje pojavljuje još i kao 'leksičko označivanje', 'morfološko označivanje' ili 'označivanje vrsta riječi'. Iako je do danas razvijeno više različitih modela označivača, čini se kako je njihova osnovna struktura zajednička svim modelima, a čine je opojavničanje, leksička analiza i uklanjanje višeznačnosti.

- Prvi korak u obradi je **opojavničanje** (engl. *tokenisation*) teksta koji se obrađuje. Opojavničavanje je proces određivanja granica lingvističkih jedinica (riječi, brojevi, znakovi interpunkcije itd.). Smatra se nisko-razinskim procesom jezične obrade i nužna je prije bilo koje druge obrade teksta. Jedinice dobivene procesom opojavničavanja nazivaju se **pojavnice**. Iako se na prvi pogled opojavničavanje može činiti

<sup>33</sup> Misli se na određivanje gramatičkih kategorija tj. u hrvatskom jeziku određivanje deset vrsta riječi (imenica, glagola, pridjeva, zamjenica, brojeva, priloga, prijedloga, veznika, čestica i uzvika).



jednostavnom procedurom u kojoj nam praznine i znakovi interpunkcije pomažu naći rubove pojava, neke druge oznake, poput crtice i apostrofa, čine taj proces nešto složenijim. Za razliku od hrvatskoga, provođenje opojavničavanja za većinu orijentalnih jezika veoma je zahtjevno jer ne postoje jasne granice između pojava. U hrvatskom jeziku pak u većini slučajeva jedna pojava pripada jednoj riječi. No, mali broj pojava koje čine više od jedne riječi je ipak dovoljan za 'stvaranje problema'. Takvi su i primjeri:

✂ *auto-cesta*  
 ✂ *je l' to dobro*  
 ✂ *1991-1995.*

- Nakon opojavničavanja obično slijedi **leksička analiza** koja uključuje primjenu flektivnoga leksikona na tekst i dodjeljivanje ispravnih oznaka. Pogađivač<sup>34</sup> se potom primjenjuje na one pojavnice koje su ostale nepoznate leksikonu. Ovim se procesom svakoj pojavnici pridružuju sve moguće oznake koje ona, na osnovi svoga oblika, može imati, npr.

✂ *KADA\_N+f+c+Noms;*  
 ✂ *KADA\_N+f+c+Genp;*  
 ✂ *KADA\_prilog;*  
 ✂ *PRIČA\_N+f+c+Noms;*  
 ✂ *PRIČA\_N+f+c+Genp;*  
 ✂ *PRIČA\_V+3+s+f+pres;*

<sup>34</sup> Pogađivač (engl. *guesser*) je program koji pogađa, odnosno na osnovi zadanih pravila, predlaže oznaku za one pojavnice koje nisu prisutne u leksikonu.

✎ *PRIČA\_V+3+s+m+pres;*

✎ *PRIČA\_V+3+s+s+pres.*

- Treća stepenica je **razobličenje** (engl. *disambiguation*) tj. razrješenje višeznačnosti koja se rješava statističkim podatkom o broju pojavljivanja određene oznake, kontekstnim informacijama u obliku pravila ili kombinacijom ove dvije metode.

✎ *Nitko nije znao kada je kada kupljena.*

✎ *Priča se kod nas priča svaku večer.*

✎ *He can can the can.*

Pojam kontekstne informacije Vanroose (Vanroose, 2001:10) opisuje kao vrstu informacije koja služi kao pomoć kod jednoznačnog određivanja vrste riječi. Ova informacija sadrži podatke o oznakama okolnih riječi, tj. riječi koje prethode i/ili slijede višeznačnoj riječi čiju višeznačnost pokušavamo riješiti.

Cutting i njegov tim (Cutting, Kupiec, Pedersen, Sibun) navode 5 svojstava koje svaki označivač mora imati:

- **robustnost** – sustav se ne sruši ako naiđe na riječ koja je označivaču nepoznata, negramatična;
- **efikasnost** – vrijeme obrade raste linearno s porastom količine teksta;
- **točnost** – pokušati svakoj riječi pridružiti točnu POS oznaku;
- **mogućnost podešavanja** – označivač se može podešavati različitim lingvističkim pomoćnim oznakama za različit korpus;
- **ponovna iskoristivost** (engl. *reusable*) – lako prilagodljiv novom korpusu, novom skupu oznaka i novom jeziku.

Može se primijetiti (vidjeti poglavlje 2.5) kako se većina svojstava za označivače odnosi i na osnovna svojstva parsera koja daje Nivre. Više o tome u poglavlju koje slijedi.

### 2.3.1.2. Povijest označivača

Najstariji i najjednostavniji označivači datiraju još iz ranih 1960-ih godina. Njihov osnovni zadatak bio je pridruživanje oznaka riječima koristeći se rječnikom. Na taj se način dobiju informacije o svakoj riječi posebno, ali i o riječima s kojima ta riječ graniči. To je od posebne važnosti za određivanje frekvencije pojavljivanja određenih konstrukcija u tekstovima.

Sljedeća generacija označivača imala je čitav niz ručno izvedenih pravila koja su dodjeljivala oznake. To se označavanje obavljalo na osnovi uzorka znakova neke riječi, ali i na osnovi riječi koje su joj prethodile ili koje su dolazile poslije nje. Leksikon koji se pri tome koristio bio je malen i sadržavao je uglavnom iznimke pravila. Jedan takav označivač bio je i TAGGIT kojeg su razvili Greene i Rubin 1971. godine. Bio je osmišljen za osnovno označavanje Brown korpusa. Sastojao se od ukupno 3 300 pravila za razobličenje i 87 oznaka za označivanje. Nakon što bi označivač prošao zadani dio korpusa, označavajući samo one riječi čije su susjedne riječi jednoznačno označene, provela bi se i ručna kontrola – odnosno ispravljanje<sup>35</sup>.

Nakon toga se, kasnih 1970-ih, pojavio CLAWS1<sup>36</sup> program<sup>37</sup> kojeg je opisao Garside, a razvijen je na Sveučilištu Lancaster za potrebe označavanja Lancaster-Oslo-Bergen korpusa. CLAWS je bio sličan TAGGIT-u ali je imao i statističko obilježje, odnosno, komponentu vjerojatnosti kojom se birala najvjerojatnija oznaka višeznačne riječi. Vjerojatnost se zasnivala na mogućnosti pojavljivanja određene oznake u danom kontekstu. Matrica koju je CLAWS1 sustav koristio, predstavljala je niz od dvije riječi, tzv. dvopojavnica (engl. *bigram*). Točnost koja se postizala bila je čak visokih 96% što je čak i kasnijim označivačima bilo teško nadmašiti.

Primjenjujući dinamičko programiranje<sup>38</sup> i ponešto usavršavajući CLAWS izdanje označivača, Steven J. DeRose se nešto kasnije (DeRose,

<sup>35</sup> Oko 77% riječi Brown korpusa uspješno je označeno dok je preostalih 23% moralo proći ljudsku provjeru.

<sup>36</sup> CLAWS = engl. *Constituent Likelihood Automatic Word-tagging System*.

<sup>37</sup> Programski kôd za CLAWS pisan je u Pascalu.

<sup>38</sup> Dinamičkim programiranjem se tablica rješenja sustavno puni rješenjima potproblema kako bi na kraju sadržavala sva rješenja potrebna za rješavanje cijelog problema.

1988.) pojavljuje sa svojim označivačem. DeRosin označivač koristi VOLSUNGA algoritam kojeg karakterizira rad u linearnom, a ne eksponencijalnom, vremenu i prostoru. Koristi 97 oznaka za označivanje koje predstavljaju modificirani oblik TAGGIT oznaka. Koristi se treniranim korpusom na osnovi kojeg gradi tablicu vjerojatnost. Testiranje provedeno na Brown korpusu postiglo je točnost od 96%.

### 2.3.1.3. Vrste označivača

Ovisno o tipu algoritma koji koristi, razlikujemo tri vrste označivača.

- Označivač može biti temeljen na pravilima što podrazumijeva veliki broj ručno napisanih pravila za uklanjanje višeznačnosti poput EngCG označivača (Karlsson, 1995).
- Drugi tip označivača temelji se na statističkim metodama koristeći pri tome trening korpus na osnovi kojeg se računa vjerojatnost određene oznake u danom kontekstu, poput HMM označivača.
- Treći tip označivača je hibridni označivač tj. kombinacija je prethodna dva jer koristi i pravila i statistiku. Takav je i Brillov označivač (Brill, 1993.).

Prvu skupinu označivača čine označivači koji koriste pravila (engl. *rule-based taggers*) i obično podrazumijevaju postojanje velike baze podataka s ručno napravljenim pravilima za rješavanje višeznačnosti (npr. riječ je imenica, a ne glagol ako ispred nje stoji član). Vanroose (Vanroose, 2001:12) vjeruje da se većina uspješnih označivača temelji na pravilima upravo zato što prirodni jezici uglavnom velikim dijelom prate, tj. zadovoljavaju, svoja gramatička pravila.

Primjer ovakvog označivača je ENGTWOL<sup>39</sup>. Ovi se označivači ne temelje na HMM modelu koji, prema Vanrooseovom mišljenju, ne predstavlja dobar izbor označivača za označavanje rečeničnih dijelova jer ne prikazuje

---

<sup>39</sup> ENGTWOL je dvo-razinski morfološki analizator za engleski jezik. Za svaku riječ u tekstu daje u prosjeku dvije alternativne morfološke oznake. Vidi više na <http://www2.lingsoft.fi/cgi-bin/engtwol>

dovoljno informacija za takvo označavanje. Kao alternativni pristup HMM-u, Atro Voutilainen predstavlja gramatiku ograničenja (engl. *Constraint Grammar*, CG) unutar koje se više oznaka pripisuje riječima na osnovi leksikona i morfološke analize.

S timom finskih lingvista, početkom 1990-ih, dizajnira EngCG<sup>40</sup> program za označavanje i plitko parsanje koji se temelji na unaprijed zadanim pravilima. Upotrebom EngCG programa, provode se kontekstualna pravila koja dovode do uklanjanja viška oznaka, odnosno oznaka koje se ne uklapaju u zadana pravila. Zabilježeno je da, iako samo pisanje pravila za uklanjanje višeznačnosti uzima dosta vremena, izvedba<sup>41</sup> ovoga modela je, ako ne ista onda, čak i nešto bolja od izvedbe HMM modela.

HMM model je posuđen iz ogranka prepoznavanja jezika i na osnovi tog model napravljen je, dosta uspješan, pokušaj njegove primjene na označivanje. Ovaj se model, zbog svoje točnosti i mogućnosti primjene nad neoznačenim tekstom, počeo uskoro smatrati standardnim označivačem. Njegova osnovna stanja su oznake ili n-torke oznaka. Svaka oznaka može imati svojstvo tranzicije i emisije. Oznaka je prijelazna, odnosno, oznaka ima prijelaznu vjerojatnost u odnosu na prethodnu oznaku, a vjerojatnost emisiju u odnosu na sama sebe, odnosno, na riječ koja se označava. Umnožak te dvije vjerojatnosti u rečenici daje vjerojatnost za određeni niz vrsta riječi. Obje vjerojatnosti se mogu izračunati primjenom algoritma naprijed-nazad (engl. *Forward-backward algorithm*), poznatim još i kao Baum-Welchov algoritam, dok se najvjerojatniji niz oznaka može dobiti primjenom algoritma Viterbi (Abney, 1996b:3).

Kod označivača koji koriste skriveni Markovljev model, svaka oznaka riječi ovisi o oznaci prethodne riječi i ta je ovisnost konstantna što znači da se ne mijenja vremenom, odnosno, ne ovisi o količini označenog teksta.

No, ipak, treba napomenuti kako se HMM označivači nisu pokazali kao dobar izbor označivača u slučajevima kada ne postoji veliki trenirani korpus,

---

<sup>40</sup> EngCG koristi čak 139 morfoloških oznaka za opis dijelova rečenice, fleksija, derivacija i nekih sintaksnih svojstava te 30 funkcionalnih oznaka za opis ovisnosti.

<sup>41</sup> Točnost EngCG modela je čak visokih 99,7% u odnosu na bolje HMM označivače čija se točnost kreće oko 95%.

ili kad se želi označiti specijalizirani tekst iz područja koje nije pokriveno treniranim korpusom, ili ako trenirani korpus uopće ne postoji.

Treću skupinu označivača predstavlja kombinacija prethodna dva, a zovu se još i transformacijski označivači (engl. *transformation-based tagger*). U ovom se slučaju korpus označava najopćenitijim pravilima, zatim specifičnim pravilima, a potom i najspecifičnijim pravilima. Primjer ovakvog označivača je Brillov označivač<sup>42</sup>. Brill je naime razvio jednu od tehnika za mehaničko usvajanje pravila iz ručno označenih tekstova za označivače koji označavaju na osnovi pravila. Označivač radi na principu da svakoj riječi pridruži najčešću tj. najfrekventniju oznaku, a sve nepoznate riječi označi kao imenicu. Ova je metoda poznata i kao Brillova metoda učenja pravila (engl. *Brill's rule-learning method*). Brillova metoda uspješno je korištena za POS označivanje, NP razdjeljivanje (engl. *chunking*), parsanje, ispravljanje grešaka u pisanju, segmentaciju i razumijevanje poruka.

### 2.3.2. Razdjelnici

Parseri prirodnoga jezika obično obrađuju tekst u dvije faze. U prvoj fazi opojavničavatelji, morfološki analizatori, prevode niz znakova u niz riječi dok u drugoj fazi sintaksni analizator ili parser prevodi niz riječi u parsanu rečenicu, tj. u niz parsanih rečenica.

Kod parsanja sintagmi, sintaksna se analiza odvija na dvjema razinama koje Abney (Abney, 1991:259) naziva razdjelnik (engl. *chunker*) i sastavljač (engl. *attacher*). Na prvoj razini dolazi do prevođenja niza riječi u niz označenih sintagmi, a na drugoj razini se niz označenih sintagmi prevodi u niz rečenica, odnosno, sintagme se sastavljaju jedna do druge dodajući one lukove koji nedostaju između čvorova parsnog stabla.

Tjong Kim Sang (2002.) opisuje razdjelnik kao cjepatelja rečenice na označene sintagme koje se sastoje od niza slijednih riječi. Te su riječi na sintaksoj razini međusobno povezane na način da se sintagme međusobno ne preklapaju i nisu rekurzivne. Sam proces razdjeljivanja dijeli se na dva

<sup>42</sup> [http://www.cs.jhu.edu/~brill/RBT1\\_14.tar.Z](http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z)

zasebna zadatka. Prvi je pronalaženje samo imenskih sintagmi (NP) a drugi je identificiranje proizvoljnih sintagmi.

Abney (Abney, 1991:261) uspoređuje razdjelnik s nedeterminističkim parserom s lijeva na desno (LR parser) koji koristi pretragu prvi najbolji (engl. *best-first search*). U razdjelniku vidi dva izvora za nedeterminizam:

- krajnje točke sintagmi nisu izravno označene u nizu riječi,
- jedna riječ može pripadati u više kategorija (npr. imenica i glagol: priča /se priča) pa rastavljač ne može odlučiti kojoj kategoriji da pridruži danu riječ.

Prema Abneyevoj definiciji (Abney, 1991:257) označena sintagma (engl. *chunk*) je jedna riječ koja nosi značenje, a okružena je konstelacijom funkcijskih riječi s kojima se uklapa po definiranom predlošku.

On smatra da je bezokolinska gramatika sasvim dovoljna za opis strukture sintagme. Odnosi koji postoje među sintagmama više su određeni leksičkim izborom nego krutim predlošcima. Zajedničko pojavljivanje sintagmi, tj. skupina riječi koje pripadaju jednoj sintagmi, određeno je koliko njihovim sintaksnim kategorijama toliko i riječima koje im prethode. Poredak označenih sintagmi mnogo je fleksibilniji od poretka riječi unutar označenih sintagmi.

Koje će se sintagme koristiti u obradi, ovisi, koliko o jeziku, toliko i o autoru koji ih primjenjuje. Tako Abney (Abney, 1996c:1) navodi 7 kategorija sintagmi za engleski jezik. U primjerima koji slijede, elementi sintagme navedeni su unutar uglatih zagrada:

- **imenska sintagma** – proteže se od početka imenske fraze do njezine glavne imenice;

✎ [little white flower]

- **glagolska sintagma** – uključuje pomoćne glagole, modalne glagole i završava s glavnim glagolom ili s glagolskim pridjevom, npr.

✎ [is fun], [is interesting], [is still difficult];

- **infinitivna sintagma** – glagoli u infinitivu uključno sa 'to';

*[to sing]*

- **sintagma prezenta participa ili gerunda** – glavni glagol koji završava na 'ing', npr.

*couldn't stop [drinking]*

*[drinking] is not good for you;*

- **pridjevska sintagma** – počinje s prilogom i 'pojačivačem' i završava s glavnim pridjevom, npr.

*[completely silent],*

*[as quiet] as a mouse;*

- **priložna sintagma** – priložne sintagme s više riječi npr.

*he ran [very quickly],*

*[three weeks later],*

*[hardly even] his mother;*

- **sintagma participa prošlog** - svi postnominalni i sekundarni glagolski pridjevi koji su morfološki participi prošli, npr.

*he is [very tired].*

Na sličan smo način odredili sintagme za hrvatski jezik koje smo detaljnije opisali u poglavlju 3.4. Te sintagme su:

- **imenska sintagma** – uključuje imenicu i sve zamjenice, brojeve i pridjeve ispred nje a koji se s njom slažu u rodu, broju i padežu;

*[jedan moj mali cvijetak]*



- **atributna sintagma** – uključuje dvije imenice od kojih je jedna atribut drugoj; sintagma uključuje i pridjeve koji pobliže označavaju navedene imenice i koji se međusobno slažu u rodu, broju i padežu;

*[velikom padu male pahulje]*

- **apozicijska sintagma** – uključuje dvije imenice od kojih je jedna apozicija drugoj i koje se međusobno slažu u padežu i broju te njihove pridjeve koji ih pobliže označavaju i s kojima se međusobno slažu u rodu, broju i padežu; u sintagmi se može nalaziti i jedna posvojna zamjenica s imenicom kao apozicijom ako se slažu u rodu, broju i padežu;

*[mali pas skitnica]*

*[naše okupljanje]*

- **glagolska sintagma** – uključuje sva jednostavna vremena i sva složena vremena, njihove negacije i povratnu česticu 'se' u slučaju povratnih glagola;

*[se ne bi bio vratio]*

- **prijedložna sintagma** – uključuje jedan prijedlog i imensku sintagmu, atributnu sintagmu ili apozicijsku sintagmu koja ga slijedi, a s kojom se slaže u padežu.

*[o tom prvom slatkom poljupcu]*

Svaka od navedenih kategorija prema Abneyu (Abney, 1996c:2) može biti navedena **parcijalno** ili **potpuno**. Tako je npr. moguće da se unutar potpune imenske sintagme nalazi parcijalna pridjevska sintagma:

*[imenski the [pridjevski new] building].*

Prijedlozi, veznici, znakovi interpunkcije i prilozi se često ne nalaze unutar sintagme osim ako su veznik ili interpunkcijski znak 'zarobljeni' unutar dijelova sintagme, npr.

✂ [the Ways and Means Committee]

✂ [the «New Delphi»].

Razdjelnik za hrvatski jezik kojeg ću prikazati u ovom radu ipak uključuje prijedlog kao glavu prijedložne sintagme, te veznike 'i', 'ili', 'ni', 'niti' u konstrukcijama kada povezuju dvije imenske sintagme koje se slažu u padežu tj. u slučajevima koordinacije.

✂ [dječaku i djevojčici]

✂ [moje kruške i tvoje jabuke]

Koristeći učenje temeljeno na transformacijama, Ramshaw i Marcus (Ramshaw, Marcus, 1995.) razvijaju razdjelnike za prepoznavanje NP i VP sintagmi. Oni predlažu i dodatne oznake za svaku riječ u rečenici po sljedećem principu:

- XB – početna riječ sintagme X
- XI – ne-početna riječ unutar sintagme X
- O – riječ koja se ne nalazi u niti jednoj sintagmi.

Na taj se način svakoj riječi, uključujući i interpunkcijskim oznakama, dodaju dodatne oznake o tome kojoj sintagmi pripadaju i na kojem se mjestu, u odnosu na sintagmu, nalaze. Mnogi su autori preuzeli ovaj model dodatnog označavanja.

Iako su i drugi autori razvili svoje modele za prepoznavanje osnovnih NP sintagmi, svi su se temeljili upravo na modelu kojeg su predložili Ramshaw i Marcus. Njihovu međusobnu usporedbu prikazali su Tjong Kim Sang i Veenstra u (Tjong Kim Sang, Veenstra, 1999.).

U *The Oxford Handbook of Computational Linguistics*, sintagma se definira kao niz riječi u tekstu koji se sastoji od osnovnog nerekurzivnog grupiranja određene sintaktičke kategorije.

Nivre (Nivre, 2006.) ga opisuje kao alternativni oblik parcijalnog parsiranja tj. kao proces identificiranja i klasificiranja nepreklapajućih<sup>43</sup> segmenata rečenice koji sačinjavaju osnovne ne-rekurzivne skupine koje odgovaraju glavnim POS oznakama (NP, VP, PP, AP). S obzirom da tekst s označenim sintagmama nema hijerarhijsku strukturu, za označavanje sintagmi se uglavnom koriste uglate zagrade s oznakom sintagme.

✎ [R Jučer] [VP sam pojeo] [NP veliki komad torte].

Dva osnovna zadatka razdjelnika su:

- pronaći nepreklapajuće skupine i
- dodijeliti svakoj skupini točnu oznaku skupine.

Koliko god se ti zadaci činili trivijalnim, u teoriji i u praksi su uslijedili različiti pristupi koji bi doveli do njihovih što točnijih rješenja. Tako su se uz razdjelnike temeljene na gramatikama, mnogi znanstvenici počeli baviti i statističkim razdjelnicima, među ostalima Church (1988.), Skut i Brants (1998.), Zhou, Su i Tey (2000.), Chen i Chen (1994.).

Skut i Brants (Skut, Brants, 1998.) su predložili stohastički razdjelnik kojim bi se prepoznavale samo sintaksne strukture određene dubine. U svom modelu oni koriste markovljev model kojim prepoznaju rubove sintagme, njezinu unutarnju strukturu i sintaksnu kategoriju. Sintagme koje su testirali su NP, PP i AP.

Chen i Chen (Chen, Chen, 1994.) daju primjer statističkog razdjelnika za određivanje NP sintagmi kojeg opisuju kao parser linerne strukture. Rezultati njihovog razdjelnika, koje je moguće automatski vrjednovati, daju visoke vrijednosti za preciznost (95%) i odziv (96%).

<sup>43</sup> U većini sustava za određivanje sintagmi, pojam *nepreklapajući* podrazumijeva da se unutar jedne sintagme ne nalazi niti jedna druga sintagma. Zbog toga na primjer, prijedložna sintagma sadrži samo jednu riječ (prijedlog) u primjeru poput [NP a car][PP from][NP London City]. Ja ću u ovom radu napraviti izuzetak ovom pravilu samo za prijedložnu sintagmu PP unutar koje ću dozvoliti pojavljivanje imenske sintagme NP *NP a car[PP from][NP London City]*.

Bes, Lamadon i Trouilleux u (Bes, Lamadon, Trouilleux, 2004.) prikazuju model razdjelnika za detektiranje glagolskih sintagmi u francuskom jeziku koji uz glavni glagol još može imati jedan od dva moguća tipa pomoćnog glagola, zamjenicu za označavanje klitike, negaciju, prilog, glagol u infinitivu i slučajnu sintagmu kao što je prikazano primjerom u kojem je VP sintagma podcrtana:

✍ *Max a, par ailleurs, mangé.*

Njihov se razdjelnik temelji na pravilima lokalnih gramatika i jednostavnim statistički svojstvima. Osnovni cilj im je bio pokazati važnost informacija za određivanje rečeničnih sintagmi, posebice VP-ova. Rezultati za preciznost i odziv njihovog sustava iznose 98%.

Uspoređujući rezultate svog modela za označivanje imenskih sintagmi u engleskome jeziku sa 6 drugih modela, Tjong Kim Sang (2002) daje sljedeći tablični prikaz s rezultatima za preciznost, odziv i f-mjeru:

	PRECIZNOST	ODZIV	F-MJERA
Kudoh i Matsumoto 2001.	94,15	94,29	94,22
Tjong Kim Sang et al. 2000.	94,18	93,55	93,86
Tjong Kim Sang 2002.	94,01	92,67	93,34
Tjong Kim Sang 2000.	93,63	92,89	93,26
Muñoz et al. 1999.	92,4	93,1	92,8
Ramshaw i Marcus 1995.	91,80	92,27	92,03
Argamon-Engelson et al. 1999.	91,6	91,6	91,6
<i>Vučković et al. 2008.</i>	<i>94,50</i>	<i>90,26</i>	<i>92,31</i>

Tablica 1  
Usporedba modela za označivanje <NP> sintagmi

Kao što se vidi iz tablice 1, prosjek za preciznost i f-mjeru svih sustava je oko 93% a za odziv 92%. Prvi model za označivanje imenskih sintagmi u hrvatskome jeziku izrađen je 2008. (Vučković, Tadić, Dovedan, 2008.). Ovaj je model u svojoj prvoj inačici dao iznimno dobre rezultate koji su uz bok rezultata dobivenih za engleski jezik.

Pojam označivanja sintagmi u literaturi nije dosljedan kod svih autora pa dok ga neki smatraju zasebnom obradom jezika (Abney), drugi pak taj termin koriste kao sinonim za parcijalno parsanje, plitko parsanje, ili pak sva tri pojma koriste u istom značenju<sup>44</sup> (Skut, Brants). U (Aït-Mokhtar, Chanod, 1997.) nalazimo i pojam 'segment' kojeg autori koriste za opisivanje sintagmi, a opisuju ga kao neprekidni niz riječi koji su sintaksno vezani međusobno ili za glavnu riječ segmenta tj. glavu segmenta.

U ovom će se radu pojmovi *označivanje sintagmi*, *parcijalno* i *plitko parsanje* koristiti zasebno u opisivanju triju različitih analiza. Za pojam 'označivanja sintagmi' preuzet ćemo već navedenu definiciju od Nivrea.

---

<sup>44</sup> Brants u (Brants, 1999.) navodi da se korak koji se provodi prije dubinske analize naziva parcijalno parsanje ili chunkiranje, dok se parcijalno parsanje koje se provodi za potrebe crpljenja i pronalaženja informacija naziva plitkim parsanjem.

## 2.4. Parseri

### 2.4.1. Povijest parsera

Kao proces koji je omogućio izradu prevoditelja, sučelja baza podataka i samo-opisnih baza podataka, parser je svoju primjenu našao i u lingvistici za analizu tekstova, analizu korpusa, strojno prevođenje i tekstualnu analizu biblijskih tekstova. Aho, Ullman, Knuth kao i mnogi drugi koji su se koristili različitim tehnikama parsanja, dali su toj proceduri teoretsku podlogu ranih 1970-ih godina. Već u sljedećem desetljeću sve su se više počeli koristiti modeli vjerojatnosti koji se primjenjuju kako za označavanje rečeničnih dijelova, tako i za parsanje i rješavanje višeznačnosti.

Prvi potpuni sustav za parsanje kreirao je 1958. godine Zellig Harris i nazvao ga projektom analize transformacija i diskursa (engl. *Transformations and Discourse Analysis Project*: TDAP). Ovaj parser se temeljio na kaskadama pretvarača konačnih stanja (engl. *cascade of finite state transducers*) pa su za Harrisa mnogi rekli da je živio 30 godina ispred svog vremena. Sustav se sastojao od 7 faza: pretraživanje riječnika, zamjena nekih gramatičkih idioma s jednom vrstom riječi; POS uklanjanje višeznačnosti uz pomoć pravila; pretvarač konačnih stanja s desna na lijevo koji se sastojao od pretvarača konačnih stanja s lijeva na desno za prepoznavanje jednostavnih imenskih sintagmi; pretvarač konačnih stanja s lijeva na desno za prepoznavanje jednostavnih prijedložnih i priložnih sintagmi; pretvarač konačnih stanja s lijeva na desno za prepoznavanje glagolskih skupina; pretvarač konačnih stanja s lijeva na desno za prepoznavanje rečenica. Rekonstrukciju ovog parsera napravili su prema originalnoj dokumentaciji Joshi i Hopely i prikazali u svom radu (Joshi, Heopely, 1996. ).

### 2.4.2. Definicije parsera

U literaturi nalazimo različite definicije parsera, odnosno različite definicije procesa parsanja. Navest ćemo samo neke od njih.

**Allen** (Allen, 1987.), na primjer, proces parsanja poistovjećuje s procesom obrade rečenice koji podrazumijeva korake koji utječu na strukturalni opis rečenice koristeći gramatičke opise lingvističke strukture, odnosno sintaksno i morfološko znanje. Dakle, potrebno je opisati način na koji se riječi spajaju od morfema i kako se te riječi dalje slažu u svrhu dobivanja valjane rečenice. Nakon ove prve faze, slijedi faza semantičke interpretacije kojom se preslikava strukturalni opis rečenice u logički oblik koji predstavlja njezino značenje neovisno o kontekstu. U trećoj fazi se provodi kontekstualna interpretacija koja preslikava sintaksne i logičke oblike u konačni prikaz efekata potrebnih za prikaz rečenice.

**Smith** (Smith, 1991:171) definira parsanje kao proces prepoznavanja rečeničnog niza i simultane izgradnje prikaza njezine strukture. Uređaj koji izvodi taj proces, bilo stvarno ili samo konceptualno, naziva parser.

**Karlsson** (Karlsson, 1995:1) pojam parsanja definira kao automatsko dodjeljivanje morfološke i sintaksne strukture, bez semantičke interpretacije, a na pisanom tekstu bilo koje duljine i kompleksnosti.

Unutar okvira konačnih stanja, **Voutilainen** (Voutilainen, 1997:285) opisuje proces parsanja na sljedeći način: prvo se uvode gramatički deskriptori koji predstavljaju alternative za vrijeme leksičke analize, a potom se eliminiraju nevaljano očitane rečenice.

**Hoard** (Hoard, 1998:208) sintaksnim parserom naziva algoritam kojim se gramatika primjenjuje na niz simbola.

**Grune i Jacobs** (Grune, Jacobs, 1998:11) opisuju parsanje, odnosno sintaksnu analizu, kao jedan od najjasnijih dijelova računalne znanosti. Prema njihovoj definiciji, parsanje je proces oblikovanja linearnog prikaza koji je u skladu s danom gramatikom. Linearni prikaz u ovako apstraktnoj definiciji nije, naravno, ograničen samo na rečenicu već se on primjenjuje i na računalni program, notni zapis muzike, radnje nekog rituala, pa čak i

uzorak za pletenje. Iz ovoga slijedi da se linearni prikaz odnosi na bilo koji linearni niz u kojem je pojavljivanje jednoga elementa uvjetovano pojavljivanjem prethodnoga elementa. Naravno, to nimalo ne pojednostavljuje proces parsanja jer je broj linearnih prikaza uglavnom beskonačan bez obzira o kojoj gramatici govorimo. Osnovni uvjet za gramatiku je da mora biti u stanju opisati strukturu svakoga prikaza. Grune i Jacobs (Grune, Jacobs, 1998:62) detaljnije opisuju parsanje prema zadanoj gramatici kao 'rekonstrukciju stabla izvođenja (engl. *production tree*) koje pokazuje kako se neki niz može izvesti iz zadane gramatike'.

U (**Jurafsky, Martin**, 2000:45) nalazimo jednu širu definiciju procesa parsanja koja uključuje kreiranje različitih vrsta struktura (morfološku, sintaksnu, semantičku, strukturu diskurs) u obliku niza, stabla ili mreže dok sintaksno parsanje definiraju kao mapiranje niza riječi u parsno stablo (Jurafski, Martin, 2000:392).

**Nederhof i Satta** (2002.) definiraju parsanje kao proces određivanja parsnih stabala temeljenih na zadanoj gramatici. Većina algoritama koji se koriste za parsanje su algoritmi prepoznavanja. Njihov je zadatak dakle, određivanje pripada li ulazni niz jeziku koji je opisan gramatikom ili automatom. S obzirom da svaki niz može imati više parsnih stabala čije izvođenje zahtjeva i vrijeme i prostor na disku, Nederhof i Satta predlažu metodu **tabularnog parsanja** također poznatog i pod nazivom navigacijsko parsanje. Ova metoda predstavlja jedan oblik dinamičkog programiranja i omogućava izvođenje djelomičnih parsnih stabala koja se pohranjuju u tablice i koja su potom dostupni različitim potpunim parsnim stablima.

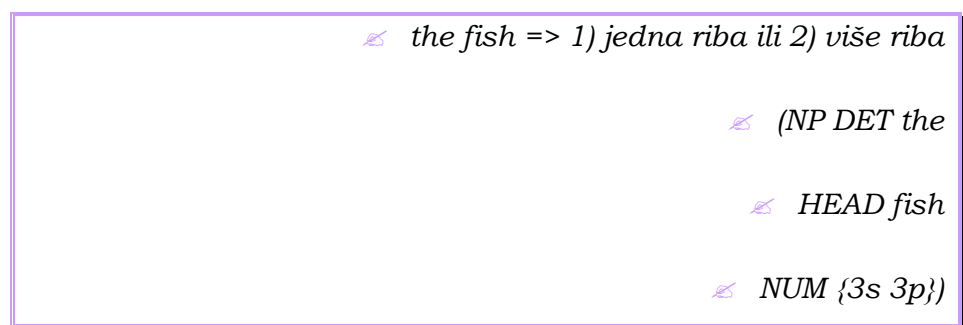
Parsanje se može opisati i kao proces izvedbe sintaksne analize nad nizom riječi. Koliko će detaljno parser opisati niz, ovisi o onome za što će se ti podaci koristiti, od izvlačenja terminoloških fraza za potrebe strojnog prevođenja, do detaljnije analize koja podržava i semantičku interpretaciju.

Još jedan od načina na koji možemo promatrati proces sintaksne analize je pretraživanje parsera kroz šumu mogućih stabala u potrazi za najboljim parsnim stablom ulazne rečenice. Područje šume koje će se pretraživati, definira se gramatikom.

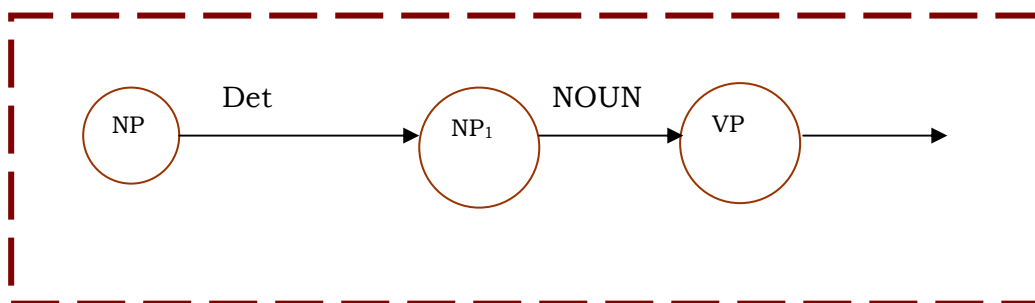


### 2.4.3. Prikaz parsera

Sintaksna struktura rečenice označava način na koji su riječi u rečenici međusobno povezane, kako se riječi grupiraju u skupine, koje riječi opisuju druge, koje su riječi od centralne važnosti u rečenici, koje veze postoje između skupina. Procesom parsanja izvode se strukturalna svojstva rečenice i daje se sintaksni prikaz kojim se pridružuje sintaksno ime svakoj osnovnoj vrsti strukture (npr. S = rečenica, NP = imenska sintagma, ...). Svojstva svake strukture identificiraju se kao skup vrijednosti. Imena svojstava, ili atributi, označavaju ulogu te vrijednosti u samoj strukturi (glavni\_glagol, objekt, subjekt, vrijeme, stanje, rod, broj, padež, lice, ...). U slučajevima višeznačnosti, sintaksni opis može uključivati listu mogućih sintaksnih prikaza kao što se vidi i iz sljedećeg primjera.



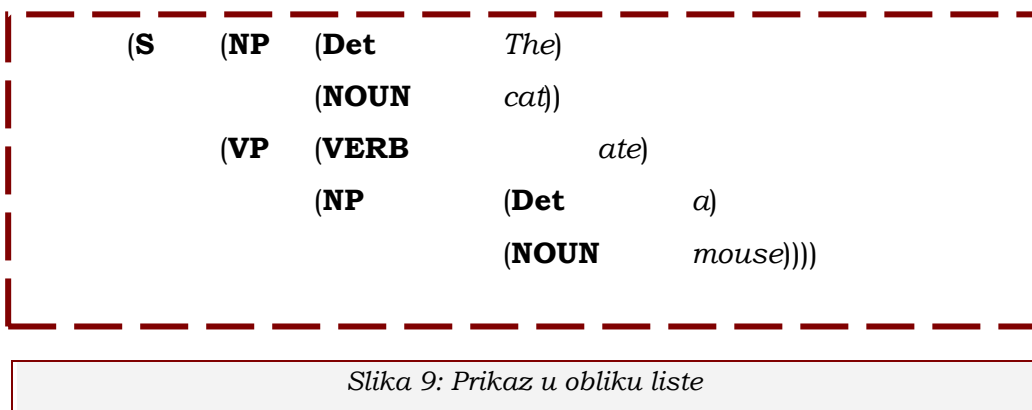
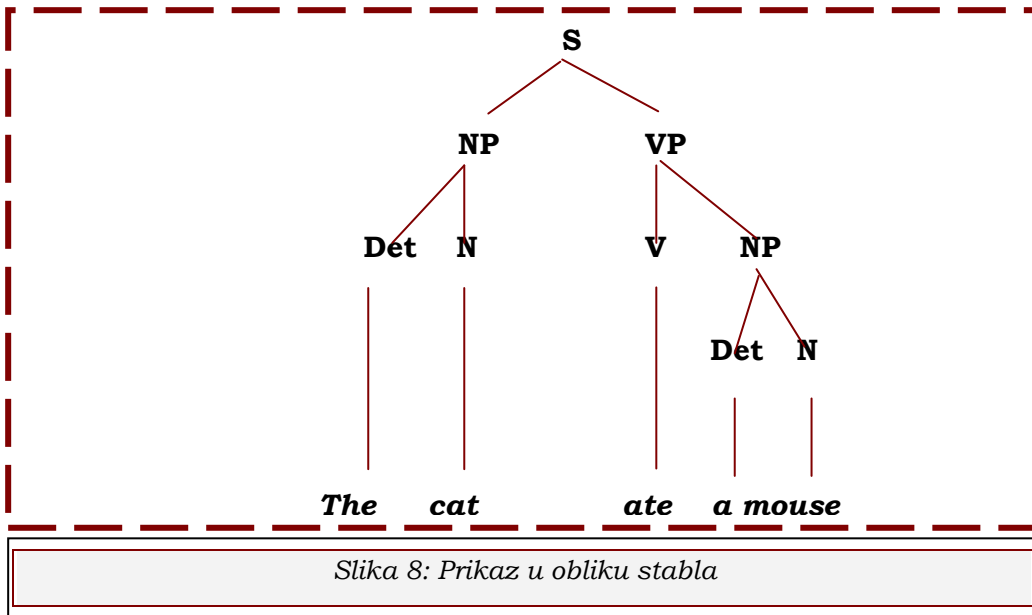
Rezultati dobiveni bilo kojom tehnikom parsanja mogu biti prikazani stablom<sup>45</sup> (Slika 8), u obliku liste (Slika 9) ili u obliku mreže prijelaza<sup>46</sup> (Slika 7).



Slika 7: Prikaz u obliku mreže prijelaza

<sup>45</sup> Parsno stablo je niz riječi sa strukturom, dok je rečenica samo niz riječi. (Grune, 1998:5)

<sup>46</sup> Mreža prijelaza se sastoji od čvorova i označenih lukova.



Prikaz u obliku stabla poznat je i kao stablični dijagram ili kao označivač sintagmi (engl. *phrase marker*) u kojem svaki čvor odgovara lijevoj strani pravila (npr.  $T1 \rightarrow T2 N1$ ; gdje su  $T1$  i  $T2$  neterminali, a  $N1$  terminal).

Stablo izvođenja potrebno je kako bi se objasnilo na koji su način i koja semantička pravila sudjelovala u stvaranju niza. Često se izbjegava prikaz stabla pa se proces parsanja prikazuje linearno i to na jedan od tri moguća načina:

- **prefiksarno** – navodi se broj čvora iza kojeg slijedi prefiksalni popis podčvorova navedenih s lijeva na desno (left-most izvođenje);

- **postfiksno** – navodi se popis svih podčvorova s lijeva na desno iz kojih slijedi broj pravila samoga nad-čvora (right-most izvođenje);
- **infiksno** – navodi se prvih  $n$  podčvorova unutar zagrada iza kojih slijedi broj pravila čvora, a potom i lista preostalih podčvorova unutar zagrada.

#### 2.4.4. Zadaci, ciljevi i karakteristike parsera

Glavni zadatak parsera je identificiranje točne sintaktičke analize među svim mogućim analizama rečenice. Što je rečenica duža, to broj mogućih analiza eksponencijalno raste. Iz toga proizlazi da je manje vremena potrebno za označiti kraće rečenice nego duže rečenice.

Smith jasno definira tri zasebna zadatka parsera i opisuje ih na sljedeći način:

- razdioba rečenice na svoje sastavne skupove, podskupove i leksičke kategorije,
- označavanje svih tih sastavnica,
- izgradnja hijerarhijskog prikaza njihovih struktura (u obliku liste ili u obliku stabla).

Abney (Abney, 1993:1-2) razlikuje dvije osnovne razine zadaće parsera: razinu rečenice i razinu izraza. Na prvoj razini određuje se vjerojatnost distribucije na skupu mogućih parsnih stabala za danu rečenicu. Na drugoj se razini svakom mogućem izrazu-kandidatu određuje vjerojatnost da pripada ispravnoj analizi. Na osnovi toga Abney zaključuje da su sve sintagme-kandidati trojke koje čine sintaktička kategorija **x**, početna pozicija sintagme **i** te završna **y** pozicija sintagme:

$$c = (x, i, y).$$

Prema Gruneu i Jacobsu, tri su razloga za provođenje parsanja:

- dobivena struktura objekta pomaže nam u njegovom još detaljnijem procesiranju;
- što je bolja gramatika koja opisuje podatke to je bolje i naše razumijevanje tih podataka;
- na osnovi postojećih podataka moguće je nadopuniti informacije koje nedostaju, ili su nepotpune, ili se pak radi o sasvim novom objektu kojeg treba označiti.

Parser treba biti moćan i fleksibilan. Nivre (Nivre, 2006:1) navodi četiri osnovna svojstva koja bi parser trebao zadovoljiti, a to su:

- **robustnost** – za svaku rečenicu u tekstu, parser bi trebao dati najmanje jednu analizu;
- **uklanjanje višeznačnosti** – za svaku rečenicu u tekstu, parser bi trebao dati najviše jednu analizu;
- **točnost** – svaka analiza koju parser ponudi trebala bi biti točna u što je moguće većem broju;
- **efikasnost** – parser bi za svaku analizu trebao koristiti što je moguće manje vremena i računalne memorije.

Kod mjerenja točnosti parsera treba uzeti u obzir vrednovanje u odnosu na ciljanu primjenu i vrednovanje u odnosu na lingvističku definiciju 'izraza'. Abney (Abney, 1996a:12) ovaj problem pokušava riješiti primjenom točnosti za lingvističko vrednovanje i iskoristivost u ciljanoj primjeni.

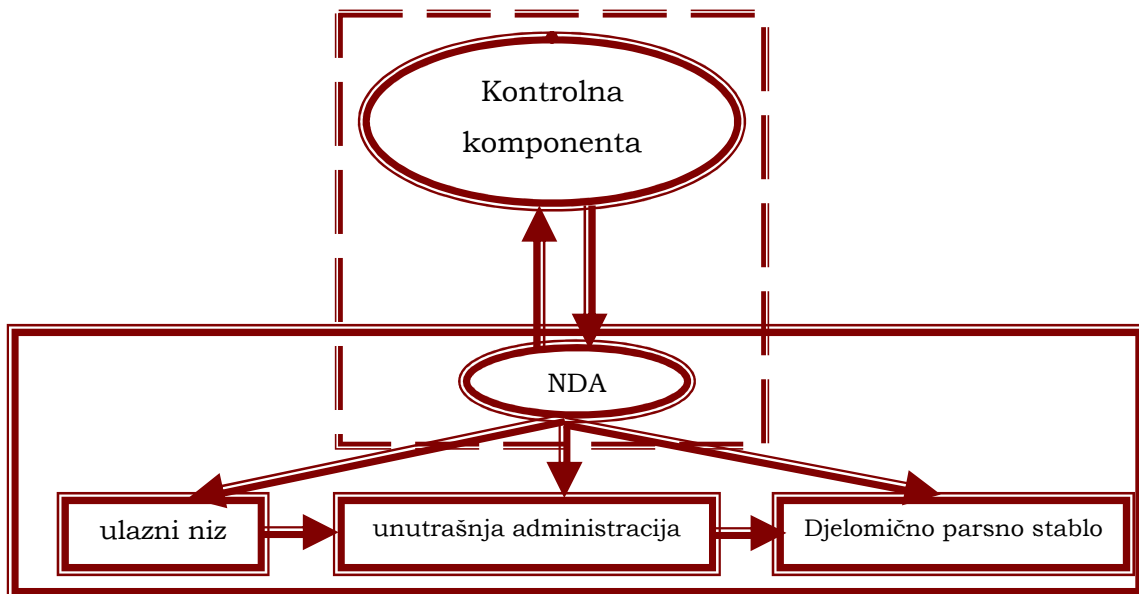
#### 2.4.5. Struktura parsera

Strukturu parsera čine dvije komponente:

- mehanizam koji vrši zamjene i bilježi parsno stablo;
- kontrolni mehanizam za provjeru koji prvoj komponenti govori koji će pomak napraviti.

Prva se komponenta još naziva i **nedeterministički automat** ili NDA jer nema unaprijed određen sljedeći potez u situacijama kada ima više

moгуćnosti. Ova je komponenta nešto jednostavnije strukture za razliku od kontrolne komponente koja je zbog velikoga gramatičkoga znanja koje mora posjedovati, dosta kompleksna.



Slika 10: Struktura parsera

Kao što se to i vidi na Slici 10, NDA je zadužen za tri stvari:

- upravljanje ulaznim nizom;
- upravljanje djelomičnim parsnim stablom;
- vođenje unutrašnje administracije.

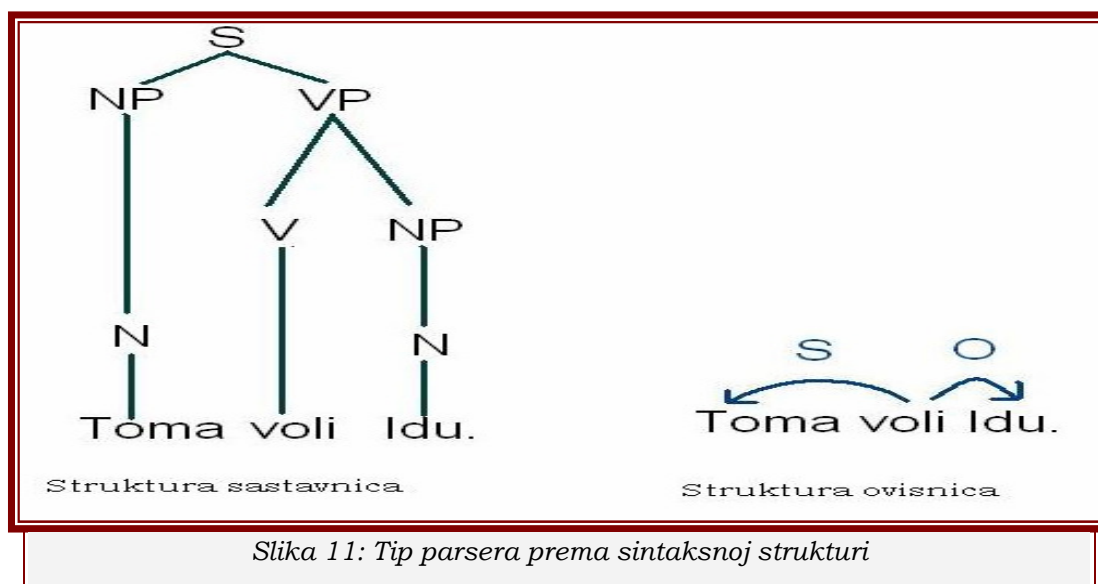
Svakim korakom NDA šalje dio informacije iz ulaznog niza preko administracije do parcijalnog parsnog stabla, ali može mijenjati i bilo koji od svoja tri modula. Važno je konstruirati takav NDA koji će, bez obzira na poteškoće s kojima se može susresti, uvijek izvršiti samo ‘valjan’ potez i tako zadržati konzistentnost svojih modula.

NDA se izvodi (kreira) direktno iz gramatike. I kod silaznog i kod uzlaznog parsera, administraciju inicijalno predstavlja početni simbol. Svi koraci koje NDA napravi u silaznom parseru u skladu su s pravilima gramatike i izvode se sve dok unutrašnja administracija nije jednaka ulaznom nizu. U uzlaznom parseru koraci su u skladu s obrnutim pravilima gramatike i izvode se sve dok unutrašnja administracija ne postane jednaka

početnom simbolu. U trenutku kad administracija poprими ciljanu vrijednost (ulazni niz ili početni simbol), parsanje je završeno.

**Kontrolne komponente** se razlikuju ovisno o tome jesu li ovisne o gramatici, konzultiraju li se s gramatikom ili ne, ili pak koriste neki tip unaprijed definiranih tablica koje su nastale na osnovi gramatike ili na osnovi ulaznog niza. Kontrolne komponente se uglavnom kreiraju programom koji se naziva generator parsera. U program se smješta gramatika i opisuju terminali, dodaju mu se jedna ili više tablica i upravljački program (engl. *driver*).

Parsno stablo, odnosno sintaksni prikaz analize, može davati pregled strukture sastavnica ili pregled strukture ovisnosti (slika 11).



Slika 11: Tip parsera prema sintaksnoj strukturi

Kao što je to vidljivo i na Slici 11, u prikazu strukture sastavnica navode se vrste riječi odnosno skupovi kojima određene riječi pripadaju, dok se u prikazu strukture ovisnosti navodi koja je riječ ovisna o nekoj drugoj riječi u rečenici. Prikaz ovisnosti u rečenici preporučuje se za jezike sa slobodnim poretком riječi a pogodan je i za pregled argumentnih struktura glagola. Prvi prikaz susreće se u gramatikama kao što su LFG, GPSG, HPSG, dok se drugom koriste *Word Grammar*, *Lexicase* i *Functional Generative Description*.

Postoji još jedan model prikaza parsnog stabla koji se sve više koristi u novijim sintaksnim teorijama poput UCG, CUG, DUG ali i LFG i HPSG. Ovaj zapis prikazuje strukturu svojstava, odnosno attribute i njihove vrijednosti pa se sukladno s tim naziva i zapis svojstava strukture (engl. *feature structure*) ili zapis atributnih vrijednosti (engl. *attribute-value*).

#### 2.4.6. Metode parsanja

Za računalnu obradu sintaksne strukture potrebno je imati GRAMATIKU tj. formalni opis dopuštenih struktura u jeziku i TEHNIKU PARSANJA odnosno metode za analizu rečenica kojima će se prepoznavati strukture ovisno o gramatici. Odabirom tehnike parsanja odabire se i način na koji se opisuje kako je gramatika proizvela rečenicu.

Tri su osnovne tehnike parsanja:

- silazna ili *top-down* metoda
- uzlazna ili *bottom-up* metoda
- kombinirana metoda.

Silazna metoda orijentirana je prema cilju tj. ona kreće od početnog simbola **S** koji je obavezni korijen za sve rečenice, pokušavajući doći do listova stabla uz pomoć postojeće gramatike. Parser na svakoj razini stabla traži pravila u gramatici čija lijeva strana odgovara trenutnoj razini te koristi desnu stranu pravila za kreiranje sljedeće tj. niže razine stabla. Kad dođe do listova stabla, vraća se i poništava ona parsna stabla kojima se nisu dobili listovi ulazne rečenice.

S druge strane, uzlazna metoda je orijentirana prema podacima tj. prema listovima stabla od kojih pokušava doći do korijena stabla. I ovdje se koriste pravila postojeće gramatike, no procedura je obrnuta od prethodne tehnike. Ovom se metodom kreće od riječi u rečenici i njihovih leksičkih kategorija koje se spajaju u skupove, koji se zatim ponovo spajaju u hijerarhijski više skupove sve dok se ne dođe do konačnog neterminala **S** tj. oznake za rečenicu. Naime, parser pridružuje svakoj riječi tj. listu stabla,

moguću POS oznaku i iz njihovih kombinacija pokušava naći identičnu desnu stranu pravila u gramatici. Ako je pronađe, kreira višu razinu stabla i pridružuje joj vrijednost lijeve strane pravila kojoj odgovara. Procedura se ponavlja sve dok se ne dođe do korijena stabla, odnosno početnog simbola **S**. Samo ona parcijalna stabla koja uspiju doći do korijena zadržavaju se kao moguća parsna stabla ulazne rečenice dok se sva ostala poništavaju. Za ovu metodu još kažemo i da je upravljana podacima (engl. *data driven*) za razliku od prethodne metode koja je upravljana krajnjim ciljem<sup>47</sup> ili hipotezom (engl. *goal driven*).

Kombinirana metoda koristi se i silaznom i uzlaznom metodom parsanja istovremeno.

Uz ovu osnovnu podjelu parsera na silazne i uzlazne, Grune i Jacobs (Grune, Jacobs, 1998:64) navode još i podjelu parsera ovisno o njihovoj usmjerenosti i o tehnici pretraživanja. Prema usmjerenosti razlikujemo **neusmjerene** (engl. *non-directional*) i **usmjerene** (engl. *directional*) metode parsanja koje mogu biti i silazne i uzlazne. Usmjerene metode obrađuju ulazni niz simbol po simbol s lijeva na desno. Glavna karakteristika neusmjerenih metoda parsanja je da parsno stablo kreiraju za vrijeme pristupanja ulaznom nizu po redoslijedu kojeg smatraju najprikladnijim za dani ulaz. Neusmjerena silazna metoda naziva se još i Ungerova<sup>48</sup> metoda dok je neusmjerena uzlazna metoda poznata i kao CKY<sup>49</sup> metoda.

Sljedeća podjela metoda parsanja je prema tehnici pretraživanja. Razlikujemo dva tipa metoda prema tehnici pretraživanja i to **pretraživanje prvo po dubini** (engl. *depth-first search*) i **pretraživanje prvo po širini** (engl. *breadth-first search*). I ove dvije metode također mogu biti i silazne i uzlazne.

Možda se na samim počecima sintaksne analize moglo govoriti o parsanju upravljanom samo gramatikom ili samo podacima, no u zadnje je vrijeme sve više kombinirano upravljanih analiza. Na ovaj se način željela poboljšati robusnost, disambiguacija, točnost i efikasnost sustava za parsanje. Modeli koji su još uvijek ostali samo u svojim čistim oblicima su

<sup>47</sup> U ovom je slučaju krajnji cilj potpuna analiza rečenice.

<sup>48</sup> Ova je metoda ime dobila po S. H. Ungeru koji ju je prvi opisao u svom članku 'A global parser for context-free phrase structure grammars' 1968. godine.

<sup>49</sup> Vidi više u poglavlju 2.5.8.



oni modeli koji koriste gramatiku ograničenja (CG) i gramatiku funkcionalnih ovisnosti (FDG) kao primjeri parsera upravljani gramatikom, dok su s druge strane parseri temeljeni na povijesti i podatkovno orijentirani parseri. Među kombinirano upravljanim parserima, odnosno parserima koji mogu biti upravljani gramatikom i/ili podacima, nalaze se parseri koji koriste leksičku funkcionalnu gramatiku (LFG), gramatike glavom upravljanih fraznih struktura (HPSG), kombinatorne kategorijalne gramatike (CCG), vjerojatnosne bezokolinske gramatike (PCFG) i gramatike rječničkih stabala (LTAG). Kombinaciju silazne i uzlazne metode parsanja predložili su Rosenkrantz, Lewis i Demers 1970-ih u modelu parsanja s lijeva.

#### 2.4.6.1. Parsanje upravljano gramatikom

Gramatika modela koji koriste metodu parsanja upravljano gramatikom može biti ručno izrađena i djelomično ili potpuno inducirana iz korpusa. Gramatike nastaju od skupova opisnih hipoteza koje se zovu pravila. Pravila trebaju biti pisana po principu Occamove britve<sup>50</sup>.

Voutilainen (Voutilainen, 1997:298) napominje kako se kod kreiranja pravila mora paziti na njegovu valjanost i efikasnost, tj. pravilo mora biti takvo da ne brani legitimne konstrukcije ali i da istovremeno smanjuje višeznačnosti.

Modeli koji se koriste ovom metodom su deterministički parseri, navigacijski parseri, parseri konačnih-stanja i parseri gramatike ograničenja. Predložena je upotreba parcijalnog parsanja za ove modele kako bi im se poboljšala robustnost i efikasnost.

**Deterministički parser** uveo je Hindle (1983.) i smatra ga se najefikasnijim parserom upravljanim gramatikom.

<sup>50</sup> Ockhamova britva (ime je dobila po engleskom franjevačkom svećeniku i logičaru Williamu od Ockhama) kojom se nastoji zadržati jednostavnost glasi: *'Entities should not be multiplied beyond necessity'*. Često se nalazi i u jednoj od dvije latinske inačice: *'Entia non sunt multiplicanda praeter necessitatem'* ili *'Pluralitas non est ponenda sine necessitate'* ([http://en.wikipedia.org/wiki/Occams\\_razor](http://en.wikipedia.org/wiki/Occams_razor)).

**Navigacijski parser** predstavio je Martin Kay (1982.) i predstavlja tip dinamičkog programiranja u kojem se uz pomoć tabličnog zapisivanja vodi evidencija o djelomičnim rezultatima.

**Parsere konačnih stanja** opisali su Abney (1996.), Koskenniemi (1997.) i Oflazer (2006.).

**Parser gramatike ograničenja** predložio je Karlsson (1995.). Ovaj parser koristi mehanizam eliminacije kojim se odbacuju analize koje ne zadovoljavaju sva ograničenja gramatike. Odbacivanje posljednje preostale analize je spriječeno i u slučaju da čak ni ona ne zadovoljava sve uvjete.

#### 2.4.6.2. Parsanje upravljano podacima

Parsanje upravljano podacima koristi banke stabala<sup>51</sup> ili parsirani korpus. Banka stabala je zbirka parsiranih rečenica željene interpretacije. Služi kao trening korpus za učenje statističkih parsera, ali i za vrednovanje kvalitete parsera. Iako je izgradnja banaka stabala zahtjevan (u vremenu i ljudima) posao, Charniak (Charniak, 1997.) smatra parsere inducirane iz banke stabala trenutno najboljim parserima. Ovisno o tome prikazuju li parsna stabla ovisnosti između sastavnica ili samo sastavnice, razlikujemo tri tipa banaka stabala:

- **banke stabala s prikazom sastavnica** poput Penn Treebank banke stabala za engleski<sup>52</sup>, arapski<sup>53</sup> ili kineski<sup>54</sup>, te Lancaster Parsed Corpus banke stabala;
- **banke stabala s prikazom ovisnosti sastavnica** poput Praške ovisnosne banke stabala za češki<sup>55</sup> i arapski<sup>56</sup>, TUT<sup>57</sup> banka stabala za talijanski, Danske ovisnosne banke stabala,

<sup>51</sup> Engleski termin *treebank* predložio je Geoffrey Leech imajući pri tome na umu stablasti prikaz sintaksne strukture rečenice. Jedna od najstarijih banki stabala je švedska banka stabala koja potiče još iz ranih 1970-ih, a sastavljena je na Sveučilištu Lund.

<sup>52</sup> <http://www.cis.upenn.edu/~treebank/>

<sup>53</sup> <http://www.ircs.upenn.edu/arabic/>

<sup>54</sup> <http://www.cis.upenn.edu/~chinese/>

<sup>55</sup> <http://ufal.mff.cuni.cz/pdt/>

<sup>56</sup> [http://ufal.mff.cuni.cz/padt/PADT\\_1.0/index.html](http://ufal.mff.cuni.cz/padt/PADT_1.0/index.html)

<sup>57</sup> <http://www.di.unito.it/~tutreeb/>

METU<sup>58</sup> banke stabala za turski, SDT<sup>59</sup> banka stabala za slovenski te HOBS<sup>60</sup> banke stabala za hrvatski (Tadić, 2007.);

- **združene banke stabala** s prikazom i sastavnica i njihovih ovisnosti poput TIGER<sup>61</sup> banke stabala za njemački i Alpino<sup>62</sup> banke podataka za nizozemski.

Osnovna podjela statističkih parsera je na one koji vrše analizu nad riječima tzv. leksikalizirani parseri i na one koji vrše analizu nad kategorijama riječi tzv. neleksikalizirani parseri. Iako leksikalizirani parseri raspolažu s više informacija, neleksikalizirani parseri su jednostavniji za napraviti zbog relativno malog skupa neterminala. No, s druge strane, neleksikalizirani parseri imaju slabije riješeno razobličenje i točnost. Statistički leksikalizirani parseri su Charniakov<sup>63</sup> model (Charniak, 1997.), Collinsov model ovisnosti i povijesno temeljeni<sup>64</sup> model kojeg je 1992. predložila Black.

Statistički neleksikalizirani parseri su parseri s djelomično nenadgledanim učenjem (engl. *Partially unsupervised learning*) kojeg su predložili Pereira i Schabes, te parsanje direktno s parsnog stabla (engl. *parsing directly from trees*).

Statistički parseri se koriste na najmanje tri načina. Jedan od njih je određivanje najveće vjerojatnosti nastavka rečenice koristeći model rešetke riječi (engl. *word lattice*). Drugi način služi za ubrzavanje samog procesa parsanja tako da se smanji broj parsnih grana. Naime, u skupu mogućih parsnih grana prvo se obrađuje statistički najčešća grana, a potom one s manjom statističkom vrijednošću pojavljivanja. Treći način je za uklanjanje višeznačnosti, tj. za svaku rečenicu s više parsnih rješenja treba odabrati prave rečenične skupine i prave ovisnosti između njih. Na taj se način dobije

<sup>58</sup> <http://www.ii.metu.edu.tr/~corpus/treebank.html>

<sup>59</sup> <http://nl.ijs.si/sdt/>

<sup>60</sup> <http://hobs.ffzg.hr/>

<sup>61</sup> <http://www.ims.uni-stuttgart.de/projekte/TIGER/>

<sup>62</sup> <http://www.let.rug.nl/~vannoord/trees/>

<sup>63</sup> Usporedbu Charniakovog, Collinsovog i Magermanovog statističkog parsera vidi u (Charniak, 1997.).

<sup>64</sup> Povijesno temeljeni (engl. *history-based*) modeli parsera mogu biti generativni i uzročni, no u oba slučaja izvode se sve moguće analize i odabire se optimalna analiza definirana probabilističkim modelom. Ovaj model radi s pretpostavkom da vjerojatnost svakog sljedećeg koraka ovisi o svim prethodnim koracima.

rang lista najboljih parsnih rješenja od kojih bi prvo na listi trebalo biti najbolje rješenje. Tako će za rečenicu

✎ *Priča se kod nas priča svaku večer.*

rang lista izgledati ovako:

✎ *<N>priča</N> se kod nas <V>priča</V> svaku večer.*

✎ *<V>priča se</V> kod nas <N>priča</N> svaku večer.*

iz čega se iščitava da bi interpretacija u kojoj je prva 'priča' imenica, a druga glagol statistički češća od druge interpretacije pa bi u rang listi došla prije svih ostalih interpretacija.

Iz opisa kojeg daje Nivre (Nivre, 2006:27) zaključujemo postojanje tri osnovne podatkovne strukture (**TIM** komponente) za metodu parsanja upravljanu podacima:

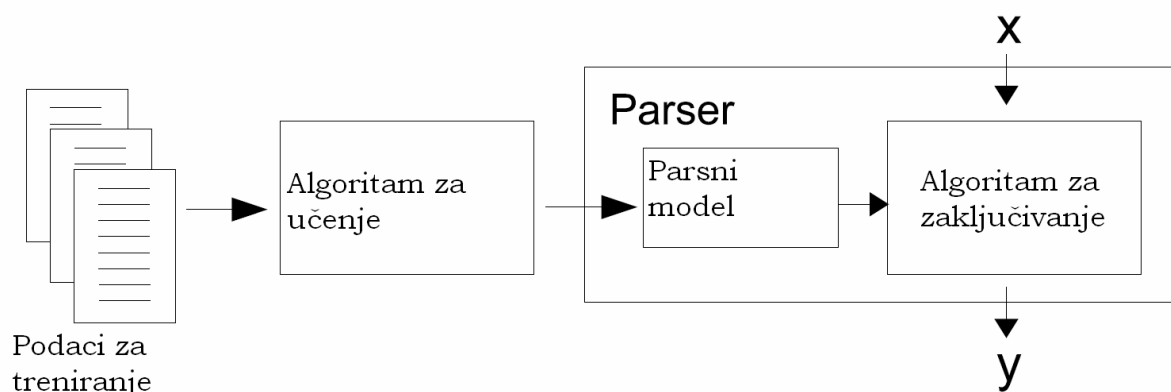
- 1. **T** – uzorak teksta iz jezika tj. **trening korpus**, koji može biti:
  - a) s točnom analizom,
  - b) bez točne analize;
- 2. **I** – **princip induktivnog zaključivanja** koji ima tri komponente:
  - a) **stohastički model** koji pridružuje vjerojatnost svakoj dopuštenoj analizi,
  - b) **parsnu metodu** kojom se računa najbolja analiza dobivena stohastičkim modelom i
  - c) **metodu učenja**, odnosno metodu zaključivanja, koja se zasniva na podacima dobivenim iz trening korpusa, a može imati oblik
    - c.1) nadgledanog strojnog učenja (engl. *supervised machine learning*),

c.2) nenadgledanog strojnog učenja<sup>65</sup> (engl. *unsupervised machine learning*);

- 3. **M – formalni model** kojim se definiraju dopuštene analize za sve rečenice koje se mogu pojaviti u jeziku.

Metoda učenja i parsna metoda induktivne komponente, međusobno su neovisne jedna o drugoj i bilo koji stohastički model može imati bilo koju kombinaciju tih dviju metoda.

Slika 12 daje grafički prikaz parsera upravljanog podacima (prilagođeno iz McDonald, Nivre, 2007.) iz kojeg se vidi da podaci za treniranje parsera ulaze u algoritam za učenje.



Slika 12: Grafički prikaz parsera upravljanog podacima

Nakon završenog procesa treniranja, odnosno učenja, kao izlaz dobivamo parser kojeg čine parsni model (tj. parametri i specifikacije svojstava) i algoritam za zaključivanje (tj. algoritam za parsanje) koji nam govori kako koristiti parsni model. Npr. za novu rečenicu *x*, koja nije bila u treniranom korpusu, algoritam za zaključivanje će, koristeći specifikacije parsnog modela, točno znati kako proizvesti sintaksni prikaz *y*.

Za metodu parsanja koriste se uglavnom CKY i Earleyev algoritam dok se za metodu učenja koristi algoritam maksimalne vjerojatnosti (engl. *maximum likelihood algoritam*) koji se može temeljiti na:

<sup>65</sup> Ovaj pristup ne zahtjeva označeni tekst. Rezultati dobiveni korištenjem algoritma unutrašnji-vanjski za nenadgledano strojno učenje, lošiji su od rezultata dobivenih za nadgledano strojno učenje.

- relativnim frekvencijama iz banke stabala,
- algoritmu unutrašnji-vanjski,
- učenju temeljenom na memoriji,
- metodi potpornih vektora,
- metodama vjerojatnosti,
- učenju temeljenom na transformacijama,
- HMM-u,
- Djelomičnom parsanju temeljenom na memoriji,
- pronalaženju gramatičkih relacija.


Kod djelomičnoga parsanja upravljanog podacima nalazimo učenje temeljeno na memoriji<sup>66</sup> i potporne vektore kao metode učenja, kod parsera upravljanog podacima nalazimo metode vjerojatnosti i učenje temeljeno na transformacijama, a kod kaskadnog djelomičnog parsanja upravljanog podacima nalazimo HMM, djelomično parsanje temeljeno na memoriji i pronalaženje gramatičkih relacija kao metode učenja.

#### **2.4.6.3. Kombinirano upravljano parsanje**

Kao primjer kombinirano upravljano parsanja navest ću model koji koristi bezokolinsku gramatiku vjerojatnosti (PCFG). Ovaj model, uz opis gramatike jezika, daje i statističku vjerojatnost svakog pravila na način da je ukupan zbroj svih vjerojatnosti, onih pravila koji imaju istu lijevu stranu, jednak 1. Npr. ako postoje dva pravila za NP od kojih prvo (pojavljivanje pridjeva i imenice) ima vjerojatnost 0,7, a drugo (pojavljivanje broja, pridjeva i imenice) ima vjerojatnost 0,3, dobijemo da je ukupan zbroj vjerojatnosti za NP lijevu stranu jednak 1. U slučaju da se pojavi još koje pravilo čija je lijeva strana NP, vrijednosti za vjerojatnost prethodna dva pravila bi se prilagodila na način da je ukupan zbroj vjerojatnosti sva tri pravila jednaka 1.

---

<sup>66</sup> Učenje temeljeno na memoriji (engl. *memory-based learning*) se u literaturi pojavljuje i kao učenje temeljeno na instancama (engl. *instance-based learning*), učenje temeljeno na primjerima (engl. *exemplar-based learning*) i učenje temeljeno na slučajevima (engl. *case-based learning*). Pojam učenja odnosi se na proces pospremanja primjera u memoriju. Na osnovi podataka u memoriji rješavaju se njima što sličniji slučajevi. Ovaj je model učenja primjenu našao u holističkom parsanju (pospremanje svih rečenica i njihovih analiza u memoriju), kaskadnom djelomičnom parsanju (zasniva se na analizi razdjelnika) i u povijsno temeljenom parsanju (Nivre, 2006:116).

 NP -> A N	p=0,7
 NP -> M A N	p=0,3

Ukupna vjerojatnost ovako dobivenog parsnog stabla dobije se zbrajanjem svih vjerojatnosti koje su sudjelovale u njegovoj izgradnji. Na ovaj se način, eliminacijom stabala s nižim ukupnim vjerojatnostima, uklanja višeznačnost.

Drugi primjer kombinirano upravljano parsanje je i induktivno ovisno parsanje (engl. *Inductive Dependency Parsing*). Ovaj parser koristi kombinaciju parsera temeljenog na gramatici ovisnosti i parsera upravljano podacima koji su potpomognuti indukcijom na nedeterminističkim točkama grananja (Nivre, 2006). Nivre ovaj sustav s tri osnovne komponente naziva MaltParser<sup>67</sup>. Čine ga **učenik**, **parser** koji koristi dvije verzije determinističkog algoritma (ORACLE-PARSE za treniranje i GUIDED-PARSE za parsanje) i **vodič** koji predviđa na osnovi svojstava ovisnosti, POS i leksičkih svojstava.

Naime, uz pomoć induktivnog učenja na osnovi podataka iz banke stabala, kreiraju se tzv. vodiči parsera kojima je osnovni zadatak predvidjeti sljedeću tranziciju na osnovu trenutnih postavku u točki grananja. Model induktivnog zaključivanja koji se ovdje koristi je iz kategorije povijesno temeljenih uvjetovanih modela (engl. *conditional history-based models*).

Srinivas, Doran i Kulick u svom radu (Srinivas, Doran, Kulick, 1995.) daju prikaz XTAG kombiniranog parsera čija su pravila temeljena na LTAG (*Lexicalized Tree Adjoining Grammar*) gramatici kojoj su dodane statističke informacije o frekventnosti korištenja svakog parsnog stabla u korpusu. Ti su statistički podaci skupljeni i pohranjeni u bazu podataka kojom se koristi parser za vrijeme analize. Drugi set statističkih podataka čine oznake prioriteta određenih pravila (npr. daj prednost da 'of' modificira NP prije nego VP, daj prednost da je 'this' determinator prije nego imenica, itd.).

<sup>67</sup> URL: <http://www.msi.vxu.se/users/nivre/research/MaltParser.html>

### 2.4.7. Prikazi parsanja

Osim metoda parsanja, u literaturi se nailazi i na podjelu parsera prema količini informacija koju prikaz nudi, te na podjelu parsera prema količini završenosti analize. Ove se podjele nikako ne smiju miješati s metodama parsanja jer se, koristeći bilo koju metodu, bilo koji dubinski i plitki parser može prikazati kao potpuno ili parcijalno parsno stablo.

#### 2.4.7.1. Količina informacije u analizi

Ovisno o količini informacije koju sintaksni prikaz analize sadrži, razlikujemo **dubinsko** (engl. *deep parsing*) i **plitko** (engl. *shallow parsing*<sup>68</sup>) parsanje. Iako je većina sustava za dubinsko parsanje upravljano gramatikama, slobodno se može reći da i dubinsko i plitko parsanje nalazimo i u metodama parsanja upravljanim gramatikama, upravljanim podacima kao i u kombinirano upravljanim metodama.

Plitko parsanje podrazumijeva djeljenje ulaznog niza na ne-preklapajuće dijelove rečenice. Svaki dio treba imati oznaku glave i opis sintaktičkih kategorija svojih sastavnica, ali ne i oznaku njihovih uloga u rečenici kao što je to slučaj s dubinskim parserima. Mnogi ga autori poistovjećuju s procesom određivanja sintagmi, tj. razdjeljivanja, dok pak drugi smatraju da je razdjeljivanje sastavni dio plitkog parsanja. Osborne (Osborne, 2000.) pak plitko parsanje opisuje kao proces osnovnog prepoznavanja sintagmi.

U praksi nalazimo različita rješenja za plitko parsanje. Tako Grefenstette (1999.) predlaže pretvarače konačnih stanja za određivanje granica imeničnih i glagolskih sintagmi, Ramshaw i Marcus (1995.) koriste statistički pristup, Abney (1996a) pak koristi regularne izraze nad oznakama vrsta riječi, dok Brants koristi Markovljev model<sup>69</sup> (1999.).

<sup>68</sup> U literaturi se još javlja i kao '*light parsing*' (vidi Grefenstette, 1999.).

<sup>69</sup> Markovljev model definira redoslijed gramatičkih kategorija, tj. što dolazi prije, a što poslije neke kategorije. Na istom je principu Andrei A. Markov 1913. definirao odnose na fonetskoj razini promatrajući redoslijed samoglasnika i suglasnika u Puškinovim djelima. Na osnovu promatranog,



Primjer plitkog modela koji koristi pisana pravila u kombinaciji sa statističkim informacijama je hibridni NP plitki parser kojeg su opisali Voutilainen i Padro (Voutilainen, Padro, 1997.) .

Daelemans, Buchholz i Veenstra (Daelemans, Buchholz, Veenstra, 1999.) opisuju plitko parsanje kao važnu komponentu u analizama tekstova za potrebe generiranja sažetaka i crpljenja obavijesti. U svom modelu predstavljaju pristup prepoznavanja NP i VP sintagmi, ali i određivanje predikat-subjekt i predikat-objekt veza.

Muñoz, Punyakanok, Roth i Zimak predlažu plitko parsanje temeljeno na modelu učenja uz pomoć rijetko rešetkaste mreže za pročišćavanje (engl. *Sparse Network of Winnows: SNoW*) koja se koristi za prepoznavanje imenskih sintagmi i subjekt-predikat sintagmi. Svoj su model testirali na dvjema paradigmama. Prva je koristila unutra/van algoritam, a druga algoritam otvoreno/zatvoreno. Prikazani rezultati pokazali su puno bolje rezultate kod korištenja druge paradigme posebno u slučajevima prepoznavanja dužih sintagmi, tj. sintagmi s više riječi.

U modelu plitkog parsanja kojeg predlažu Molina i Pla (Molina, Pla, 2002.) koristi se HMM statistički pristup za metodu učenja. U usporedbi s ostalim korištenim metodama učenja, njihov je model dao slične rezultate a pokazao se i kao efikasan i jezično neovisan.

Još jedan primjer plitkog parsanja temeljenog na podacima, i to za švedski jezik, prikazala je u svom radu Beáta Megyesi (2002.). Iste godine i Erik F. Tjong Kim Sang objavljuje rad u kojem prikazuje metodu plitkog parsanja kombiniranu s učenjem temeljenim na memoriji<sup>70</sup>. Svrha ovog modela je identificiranje osnovnih imenskih sintagmi, prepoznavanje sintagmi proizvoljnog tipa, pronalaženje surečenica, otkrivanje ugniježđenih imenskih sintagmi i puno parsanje.

Model plitkog parsera temeljenog na konačnim stanjima za francuski jezik opisali su Aït-Mokhtar i Chanod (Aït-Mokhtar, Chanod, 1997.). Već u svojoj prvoj inačici, ovaj je model pokazao jako dobre rezultate pri obradi

---

mogao je izračunati vjerojatnost pojavljivanja suglasnika ovisno o jednom ili dva slova koja mu prethode. Zbog svoje uspješnosti, model je preuzet i na višim razinama obrade jezika.

<sup>70</sup> Sustavi koji učenje temelje na memoriji klasificiraju podatke ovisno o njihovoj sličnosti s već viđenim podacima.

tehničkih priručnika i novinskih članaka. Modularnost ovoga sustava dopušta mu jednostavnu prilagodbu kako različitim vrstama tekstova tako i različitim jezicima.

#### 2.4.7.2. Potpunost analize

Podjela parsera na **potpune** i **djelomične** parsere temelji se na završenosti ili potpunosti provedene sintaksne analize i definira se u odnosu na tip prikaza koji se koristio, tj. ovisno o tome radi li se o analizi sastavnica ili ovisnica. Prema tom principu, proces određivanja osnovnih sintagmi u rečenici smatra se djelomičnim parsanjem sastavnica dok se određivanje gramatičkih funkcija svake pojedine pojavnice smatra djelomičnim parsanjem ovisnica.

Djelomično parsanje, za razliku od običnih parsera kojima je cilj producirati potpuno parsno stablo, prednost daje učinkovitom i pouzdanom prepoznavanju sintaksnih odnosa u neograničenom tekstu, pridajući pri tome manju pozornost potpunosti i dubini analize. Ovisno o svojoj složenosti, jednostavniji djelomični parseri imaju mogućnost pronaći samo određene skupine u rečenici, recimo NP-ove, dok kompleksniji djelomični parseri mogu odrediti i gramatičke funkcije skupina kao što su za NP-ove subjekt, izravni ili neizravni objekt, i slično. Najsloženiji djelomični parseri mogu još navesti i djelomične informacije o vezi dvaju susjednih skupova. Tako je u sljedećem primjeru prvi NP vezan sa sljedećim NP-om.

✎ [NP plava košulja] [NP moga brata]...

Abney (Abney, 1996b:1) opisuje djelomično parsanje kao niz različitih tehnika za otkrivanje nekih, ali ne svih informacija koje se nalaze u tradicionalnoj sintaksoj analizi.

Jedan od starijih, ali ne i manje poznatih, djelomičnih parsera je Fidditch. Hindle, autor ovoga parsera, posebno ga je namijenio neograničenim tekstovima (engl. *unrestricted text*). Fidditch se temelji na

Marcusovom parseru koji mu je prethodio i odlikuje ga velika brzina<sup>71</sup>. Ovaj vremenski mlađi parser karakteriziraju dvije bitne osobine koje njegov stariji brat nema, a to su:

- jednostavniji formalizmi,
- nova metoda – *punt*<sup>72</sup>.

Jednostavniji formalizmi omogućili su pisanje mnogo većih gramatika. Novom *punt* metodom parsanja rečenični dijelovi, čija se uloga ne može jednoznačno odrediti, ignoriraju se, a parsanje se nastavlja kao da taj dio niti ne postoji u rečenici. S obzirom na vrijeme nastanka, ovaj pristup parcijalnom parsanju ne koristi označivač vrsta riječi<sup>73</sup>.

Abney je (Abney, 1990.) kreirao deterministički parser nazvan CASS, (engl. *Cascaded Analysis of Syntactic Structure*), sličan Fidditchu, koji se sastoji od 3 osnovna filtra:

1. **filtrar sintagmi** – određuje sintagme,
  2. **rečenični filtrar** – određuje početak i kraj jednostavne rečenice te joj označava subjekt i predikat,
  3. **parsni filtrar** – slaže označene sintagme u potpuna parsna stabla.
- Svaki od tih filtara se dalje sastoji od dodatnih podfiltara zaduženih za još jednostavnije transformacije.

Djelomično parsanje se koristi za crpljenje obavijesti (engl. *information extraction*) i pronalaženje obavijesti (eng. *information retrieval*), a veliku je primjenu našlo i kod sustava za razumijevanje poruka koji prolaze kroz sljedeće korake:

- filtriranje (izdvajanje) nebitnoga teksta;
- opojavničenje<sup>74</sup>;
- parsanje temeljeno oko ključnih riječi;
- ispunjavanje semantičkih okvira;

<sup>71</sup> Abney (Abney, 1996b:10) ga smatra čak najbržim parserom.

<sup>72</sup> *Punt* metooda označava namjerno odustajanje od rješenja na način da se neki korak ipak napravi.

<sup>73</sup> Iako su ranih 1980-ih, u vrijeme nastanka Fidditcha, označivači vrsta riječi već postojali, još uvijek nisu bili toliko pristupačni.

<sup>74</sup> Segmentiranja ulaznog niza u uređeni niz pojava (engl. *token*) od kojih svaki odgovara jednom tipu simbola (broj, interpunkcija, oblik riječi, itd.) koji se dalje šalje na obradu. Opojavničavatelj je deterministički ako se izlazni niz sastoji od samo jednoznačnih dijelova, tj. ne postoji alternativna segmentacija.

- spajanje okvira u svrhu upotpunjavanja podataka predložaka.

Nakon provođenja djelomičnoga parsanja, pristupa se spajanju sintaksnih dijelova. To je spajanje omogućeno preko semantičkih okvira koji se zbog praktičnosti trebaju napraviti ručno. Semantički okvir glave određuje semantički okvir označene sintagme pa dvije sintagme mogu biti spojene samo ako semantički okvir prve može popuniti utor semantičkoga okvira druge sintagme.

#### **2.4.8. Algoritmi korišteni u parsanju**

Najznačajniji algoritmi koji su se koristili do sada za potrebe sintaksne analize su: EM – maksimizacija očekivanja, Baum-Welch, Viterbi, Earley, CKY, algoritam lijevog kraja i ulazno-vanjski algoritam. Ovdje ćemo ih ukratko opisati.

##### **2.4.8.1. Earleyjev algoritam**

Earleyjev algoritam predložio je u svom doktoratu J. Earley još 1968. godine i to za potrebe opisa postava na formalnim jezicima. U godinama koje su uslijedile, algoritam je preuzet i za potrebe obradbe sintakse prirodnoga jezika te je u međuvremenu dobio više svojih inačica i poboljšanja od kojih ćemo ovdje samo neka spomenuti.

Earleyjev se algoritam također ubraja i u algoritme dinamičkog programiranja, ali i u algoritme navigacijskog parsanja. Za razliku od CKY algoritma, Earleyjev algoritam koristi silaznu tehniku obrade.

U jednostrukom prolazu slijeva, ovaj algoritam ispunjava mjesta u polju sa stanjima koja prikazuju do tada generirana djelomična parsna stabla. Svako stanje može se obraditi na jedan od tri moguća načina: skeniranjem, upotpunjavanjem ili predviđanjem. Skeniranjem se dodaje stanje sljedećem ulazu u polju dok se upotpunjavanjem i predviđanjem dodaju stanja trenutnom ulazu koji se obrađuje. Najprije predviđač (engl.

*predictor*) dodijeli trenutnom ulazu sva moguća stanja koja su dopuštena iz trenutnog neterminala. Zatim skener (engl. *scanner*) provjerava može li se ulaz s predviđenom oznakom smjestiti u dobivenu analizu tj. postoji li terminal s predviđenom oznakom. Ako je odgovor potvrđan, kreira se novo stanje u sljedećem polju. Kada se dođe do kraja desnog pravila, poziva se upotpunjavač (engl. *completer*) kojim se određuje da je uspješno prepoznata, odnosno prihvaćena, cijela desna strana pravila. Formalni opis algoritma vidi u Dovedan (2003:97).

Za potrebe statističkih parsera, razvijene su i statističke varijante Earleyjevog kao i CKY algoritma.

Earleyjev parser kreira podatke kao vezane sekvence stanja koje predstavljaju sve moguće analize ulaznoga niza za što koristi mnogo resursa i vremena. McLean i Horspool kreirali su bržu verziju Earleyjeva parsera koji su kombinirali s LR( $k^{75}$ ) parserom te tako dobili LRE( $k$ ) parser koji je znatno brži (čak 10 do 15 puta), a pri tom koristi pola manje računalnih resursa (McLean, Horspool, 1996.).

#### 2.4.8.2. CKY algoritam

CKY<sup>76</sup> metoda ime je dobila prema trojici najpoznatijih autora, Cocke, Kasami i Younger, koji su ju predložili. CKY algoritam pripada algoritmima dinamičkoga programiranja koji koristi uzlaznu metodu. Ovaj algoritam zahtijeva bezokolinske gramatike pisane u Chomskyjevoj normalnoj formi (CNF) što znači da svaki neterminal može biti opisan ili kao jedan terminal ili kao dva neterminala.

Ovakva gramatika dopušta stvaranje dvodimenzionalne tablice. U svakom polju tablice označenom s  $[x,y]$  koordinatama koje upućuju na praznine između riječi ulaznog niza, nalaze se neterminali koji predstavljaju sve sastavnice koje se nalaze na mjestima od  $x$  do  $y$ . Takva parsna tablica ispunjava se s lijeva na desno i odozdo prema gore. Najduže vrijeme

<sup>75</sup> ( $k$ ) označava broj oznaka ispred trenutne oznake koje parser treba 'pogledati' prije donošenja odluke o analizi trenutne riječi.

<sup>76</sup> U literaturi se još pojavljuje i kao CYK algoritam.

potrebno za CKY algoritam da provede analizu nad tekstem je ( $n^3$ ) gdje je  $n$  duljina analiziranog teksta iskazana u terminalima. Formalni opis algoritma vidi u Dovedan (2003:94).

#### 2.4.8.3. EM algoritam

EM – algoritam maksimizacije očekivanja koristi se u statističkim analizama, stoga je i parser temeljen na EM algoritmu (engl. *expectation-maximization*) vrsta statističkog parsera. Koristi se za pronalaženje najvjerojatnijeg parsnog stabla koje je dobiveno izračunom vjerojatnosti svih stabala kandidata. Kandidat s najvećom vjerojatnosti uzima se kao točno parsno stablo.

Takav statistički parser je i PLTIG (*Probabilistic Lexicalized Tree Insertion Grammar*) parser. Dva poznatija EM algoritma su *Ulazno-vanjski* algoritam i grafički EM algoritam. Njihovu usporedbu prikazao je Tanaka u svom radu (Tanaka, 2003.).

EM algoritam je koristio i Dębowski (Dębowski, 2007.) za određivanje valencije glagola za poljski jezik. Algoritam je koristio za izvođenje nenadgledano razrješivanje alternativnih valencijskih okvira.

Magerman (Magerman, 1994.) EM algoritam koristi u svom statističkom parseru SPATTER<sup>77</sup> prvo za treniranje skrivenih modela izvođenja kojima pridružuje težinu izvođenja kako bi se maksimizirala ukupna vjerojatnost korpusa, nakon čega primjenjuje drugi EM algoritam kako bi poboljšao prethodno dobiveni rezultat trenirajući skrivene parametre unutar parsnih stabala.

---

<sup>77</sup> SPATTER = **S**tatistical **PATTE**rn **R**ecognizer, statistički prepoznavatelj oblika. Ovaj parser koristi modele vjerojatnosti kako bi prepoznao POS oznake, grane parsnog stabla i oznake sastavnica svake rečenice. Parser koristi uzlaznu tehniku parsanja, te EM algoritam i povijesno temeljenu gramatiku.

#### 2.4.8.4. Ulazno-vanjski algoritam

Ulazno-vanjski algoritam (engl. *inside-outside*) predstavio je 1979. James K. Baker. Predstavlja poseban slučaj EM algoritma. Kao i EM algoritam ima dva koraka. Prvi je korak očekivanja, a drugi korak maksimizacije. Koristi se za procjenjivanje vjerojatnosti nekog pravila iz gramatike koja se koristi u procesu parsanja, ali i za induciranje gramatičkih pravila. Osnovna pretpostavka ovog algoritma je da je najbolja ona gramatika na temelju koje rečenice u treniranom korpusu imaju najveću vjerojatnost pojavljivanja.



James K.  
Baker

Algoritam koristi dinamičko programiranje za brzi izračun očekivanog broja pojavljivanja nekog pravila u danom korpusu pronalazeći istovremeno sve parsne analize za niz riječi. Unutrašnja i vanjska vjerojatnost računaju se zasebno. Od tuda i sam naziv: Ulazno-vanjski algoritam.

#### 2.4.8.5. Baum-Welch algoritam



Lloyd R. Welch

Ovaj je algoritam ime dobio po autorima Leonardu E. Baumu i Lloyd R. Welchu. Radi na principu izračuna vjerojatnosti svakog izvođenja. Koristi se za optimizaciju vjerojatnosti.

U literaturi je još poznat i kao algoritam **naprijed-nazad**. Koristi se u modelima za prepoznavanje govora, za označivanje vrsta riječi ali i za parsanje.

#### 2.4.8.6. Viterbijev algoritam



Andrew Viterbi

Ovaj je algoritam 1967. godine osmislio Andrew Viterbi i po njemu je dobio ime. Viterbijev se algoritam obično koristi za proces dekodiranja, odnosno za otkrivanje niza skrivenih stanja na osnovi do tada promatranog niza. Smatra se standardnim algoritmom u dinamičkom programiranju.

#### 2.4.8.7. Algoritam lijevog kraja

Algoritam lijevog kraja (engl. *Left-corner algoritam: LC*) su 1970. predstavili autori Rosenkrantz i Lewis. Ovaj algoritam koristi uzlaznu metodu analize počevši od lijevog kraja pravila, ali istovremeno i silaznu metodu za predviđanje ostatka pravila. Naime, analiza se počinje od ciljane kategorije, odnosno korijena grane koja se treba napraviti. Iz tog se mjesta gleda u lijevi kraj pravila. U slučaju da lijevi kraj odgovara ciljanoj kategoriji, analiza je završena. U suprotnom, analiza se nastavlja projekcijom mogućeg lokalnog stabla iz lijevoga kraja. Preostala djeca novonastalog lokalnog stabla postaju ciljana kategorija pa se rekurzivno ponavlja procedura iz lijevog kraja. Po završetku kreiranja lokalnog stabla, ono postaje lijevi kraj pravila pa se ponavlja parsanje iz lijevoga kraja.

Istraživanje koje su proveli Roark i Johnson (Roark, Johnson, 1999.), pokazala su da parseri koji koriste algoritam lijevog kraja daju bolje rezultate u točnosti nego parseri koji koriste samo uzlazno ili silazno parsanje.

Moore je u svom radu iz 2004. predstavio poboljšanu verziju algoritma lijevog kraja koju je usporedio i s drugim prijedlozima istog algoritma, ali i s CKY, Earley i GLR<sup>78</sup> algoritmima. Mooreova verzija pokazala je znatno bolje rezultate od svih ostalih algoritama s kojima se uspoređivala. Razlog tome Moore nalazi u redoslijedu primjene uzlazne i silazne metode. Naime, njegovo istraživanje pokazalo je da je algoritam lijevog kraja značajno brži ako se

<sup>78</sup> GLR je kratica za generalizirano naj-desnije izvođenje s lijeva (engl. *Generalized Left-to-right Rightmost derivation*).



najprije provodi uzlazna provjera simbola koji je prvi na desno od trenutnog simbola, a tek potom silazna provjera, nego obrnuto. Na ovaj se način štedi na vremenu jer u slučaju da uzlazna metoda ne da pozitivne rezultate, silaznu metodu tada nije ni potrebno provoditi.

**Generalizirani algoritam lijevog kraja**, kojeg je 1977. godine predložio Demers, koristi uzlaznu metodu za prepoznavanje ulaznog niza sve do trenutka dok nije jednoznačno određeno s kojim se gramatičkim pravilom slaže. Nakon toga primjenjuje silaznu metodu za slaganje s ostatkom pravila desne strane. Nederhof je usporedio generalizirani algoritam lijevog kraja s generaliziranim algoritmom s lijeva i dobio puno bolje rezultate obrade (Nederhof, 1993.) pa ga zbog prikazanih prednosti predlaže kao alternativu generaliziranom algoritmu s lijeva.

Statističku verziju parsera koji koristi algoritam lijevog kraja predložili su i opisali Manning i Carpenter (1997.). I njihov model koristi simultano uzlaznu metodu koja analizira od lijevog ruba pravila i silaznu metodu koja analizira od ciljane kategorije. Statistički dio modela pripisuje vjerojatnost svakom parsnom koraku koji polazi iz lijevog kraja, a koja ovisi o trenutnoj ciljanoj kategoriji (npr. vjerojatnost ekspanzije NP bit će različita ovisno o tome je li ciljana kategorija za NP subjekt ili objekt rečenice).

#### **2.4.8.8. LR - algoritam s lijeva**

LR – algoritam s lijeva opisao je 1965. Knuth. LR parser je uzlazni parser koji radi analizu rečenice s lijeva na desno (L), a provodi izvođenje po principu zamjene najprije onog neterminala koji je prvi zdesna (R). Parsanje s lijeva na desno započinje odabirom ciljane kategorije i nastavlja se provjerom pripadnosti njezine lijeve grane. Ako obe riječi pripadaju istoj kategoriji tj. kategorija lijeve grane je ista kao i ciljana kategorija, proces se zaustavlja. U suprotnom, kreira se moguće lokalno stablo iz lijeve grane takvo da odgovara postojećem pravilu u gramatici u kojem je kategorija lijeve grana prva stavka desne strane. Npr. ako je kategorija lijeve grane A, moguće pravilo je  $NP \rightarrow AN$ . Nastavlja se s određivanjem ostale djece ovog lokalnog

stabla i tek kad se definiraju sva djeca, vraća se na određivanje ostalih lokalnih stabala glavnog parsnog stabla. Formalni opis algoritma vidi u Dovedan (2003:117).

Horspool i Whitney (1990.) daju prijedlog bržeg LR parsera koji kompajlira sve postojeće tablice u direktno izvršiv kôd te uz optimizaciju povećava brzinu parsanja (čak pet do osam puta) i smanjuje veličinu parsera.

Striktno parsanje s lijeva podrazumijeva da se na svakom mjestu u rečenici računa vjerojatnost sljedeće riječi s obzirom na prethodne riječi i nikad se ne ide unatrag radi modificiranja vjerojatnosti.

U literaturi nalazimo i jednostavni LR parser ili SLR (engl. *Simple LR parser*) zatim LALR parser koji označava LR parser s mogućnošću gledanja unaprijed (engl. *Lookahead LR parser*) te kanonski LR parser. Svaki od njih, prema navedenom poretku, može se nositi s većim brojem gramatika od prethodnog parsera. Dakle, SLR se može nositi s većim brojem gramatika od LR, LALR od SLR, a kanonski LR od LALR parsera. Dok LR parser ne koristi metodu gledanja unaprijed tj. ona u obzir uzima 0 konteksta, to nije tako za preostale parsere. LALR parseri su specifični i po tome što uzimaju u obzir više konteksta od SLR parsera.

#### 2.4.8.9. Generalizirani algoritam s lijeva



Masaru Tomita

Generalizirani algoritam s lijeva ili GLR uveo je 1985. Masaru Tomita u svojoj doktorskoj disertaciji. Parser koji ga koristi naziva se GLR parser (engl. *Generalized Left-to-Right Rightmost derivation parser*) ili Tomita parser.

Tomita parser simulira paralelnu izvedbu većeg broja kopija LR parsera i koristi DAG<sup>79</sup> strukturu podataka za smanjivanje potrebnog prostora za sve parsne stogove. Osnovni mu je zadatak provesti što potpuniju i učinkovitiju analizu rješavajući pri tome i višeznačnosti i nedeterminizam. Ovaj algoritam radi slično kao i LR algoritam iz kojeg je i

<sup>79</sup> DAG = Directed Acyclic Graph – usmjereni nepovratni graf (UNG)

razvijen, no s tom razlikom da će GLR analizirati sve moguće interpretacije po principu pretrage u širinu, uzevši u obzir zadanu gramatiku. Ovaj je parser posebno efikasan u slučajevima malog broja konflikata unutar LR stanja (McLean, Horspool, 1996).

Kao i kod CKY i Earleyjevog algoritma, vrijeme potrebnu za analizu je  $n^3$  u slučajevima kad je gramatika nedeterministička. U suprotnom, dakle, ako se radi o determinističkoj gramatici, potrebno vrijeme za analizu se smanjuje na čak  $n$ . Bržu verziju GLR parsera za višeznačne gramatike razvili su Aycok, Horspool, Janoušek, Melichar i izložili u svom radu iz 2001. GLR ili Tomita parser još je poznat i pod nazivom 'paralelni parser'.

Statistički GLR model opisali su Inui, Sornlertlamvanich, Tanaka, Tokunaga (1997.). Njihov se model temelji na GLR algoritmu uz dodatni modul za vjerojatnost.

#### 2.4.8.10. Algoritam s povratnim izvršenjem

*Backtracking*<sup>80</sup> ili algoritam s povratnim izvršenjem omogućava čuvanje zapisa odabira putanja kojima se još nije prošlo za slučaj da se mora vratiti i odabrati neku novu putanju jer prethodne nisu do kraja opisale traženi niz riječi. To omogućava ispravljanje krive interpretacije kao npr. u rečenici:

✂ *Dijete je brzo trčalo.*

ako je izraz '*je brzo*' u prvom prolazu označen kao imenski predikat, a nakon toga algoritam nađe još glagol '*trčalo*', mora se vratiti i označiti '*je*' kao pomoćni glagol, '*brzo*' kao prilog, te '*trčalo*' kao glavni glagol.

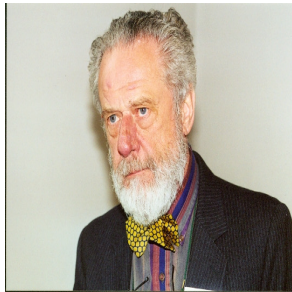
Slično je i u rečenici:

✂ *Perce je lagano lepršalo.*

<sup>80</sup> Backtracking = backward execution algorithm => algoritam s povratnim izvršenjem.

Poseban model backtracking algoritma je kronološki backtracking algoritam koji čuva zapis onih putanja koje još nisu isprobane. U slučaju da se mora vratiti, gleda kronološki zadnje mjesto izbora i tamo se vraća. Ako ni odabirom te putanje ne uspije izvršiti analizu do kraja, ponovo se vraća na mjesto izbora kronološki neposredno prije posljednjeg mjesta izbora. Iako ova metoda ne troši puno memorije za izvršenje, vremenski može dosta dugo trajati jer neke putanje prolazi više puta.

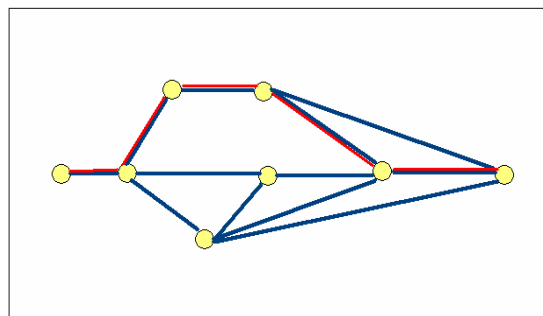
#### 2.4.8.11. Algoritam navigacijskog parsera



Martin Kay

Kao odgovor na upravo taj problem s vremenom, predstavljen je model navigacijskog parsera (engl. *chart parsing*<sup>81</sup>) koji, za razliku od prethodno opisanog modela, vodi evidenciju o već provjerenim, tj. prepoznatim strukturama. Na svakoj poziciji u rečenici postoji zapis o ispravnim (provjerenim) skupinama koje počinju i završavaju na toj poziciji što sprječava ponovnu analizu već analiziranih skupina riječi.

Model je osmislio Martin Kay, a ime je dobio po tome što način vođenja evidencije izgleda kao mapa (vidi Sliku 13) na kojoj vrhovi (žuto označeni krugovi) predstavljaju pojedinačne strukture rečenice između kojih se nalaze lukovi (plavo označeni) koji predstavljaju pravila.



Slika 13: Navigacijski parser

<sup>81</sup> U literaturi se još pojavljuje i kao algoritam tabličnoga parsiranja (engl. *tabular parsing algorithm*).

Ako je pravilo između dvije strukture bilo ispravno odabrano kreira se novi luk (crveno označen) između ta dva vrha kako bi označio 'sigurnu putanju'. I ovaj algoritam pripada grupi algoritama za dinamičko programiranje.

#### 2.4.8.12. Algoritam kaskadnog parsera



Steven Abney

Steven Abney (Abney, 1996a:8) predstavlja kaskade konačnih stanja<sup>82</sup> (engl. *finite state cascades*) kao pristupačniju arhitekturu za parsanje tekstova bez ograničenja. Upravo se na njima temelji i tehnika koju koristi za parsanje označenih sintagmi.

Kaskade konačnih stanja sastoje se od više razina tako da se nizovi jedne razine nadograđuju na nizove prethodne razine imajući na umu da niti jedan niz ne sadrži niz iste ili više razine. Dvije od tih razine Abney smatra posebnima:

- razina sintagmi – nerekurzivne sastavnice osnovnih nizova, tj. NP, VP, PP, AP, ADVP;
- razina jednostavne surečenice – surečenica čija je umetnuta surečenica pretvorena u njenu srodnu surečenicu.

Kaskade konačnih stanja predstavljaju brzu tehniku bez globalnih optimizacija što je uzrok brzine i izdržljivosti (robusnosti). Naime, zajednički problem tradicionalnih parsera je odbacivanje valjanih izraza niže razine jer se ne mogu smjestiti u globalni parser zbog neizbježne nepotpunosti gramatike. No, ovo se može izbjeći kada se takve vrste izraza procjenjuju zasebno.

Kaskade konačnih stanja imaju više razina obrade. Na svakoj od razina kreiraju se sintagme na osnovu sintagmi kreiranih na prethodnim razinama. Za prepoznavanje sintagmi koristi se automat konačnih stanja kojeg Abney naziva prepoznavać razine (engl. *level recognizer*). Njegov je

<sup>82</sup> Abney (Abney, 1996.) predlaže ovu tehniku kako bi se dobio brz i izdržljiv parser. Ova se tehnika koristi u projektima za induciranje potkategorizacijskih okvira i odabir ograničenja za engleski i njemački jezik.

zadatak da krene od početnog simbola i ovisno o riječima na koje naiđe, kreira izlaznu oznaku za taj niz po modelu najdužeg poklapanja (engl. *longest match*). U slučaju da dođe do neprepoznavanja riječi prije kraja niza, neprepoznata riječ se preskače u obradi tzv. *punt* metodom. Prepoznavanje se nastavlja na prvom sljedećem simbolu (vidi primjer u Abney, 1996a:2). Ovakav model parsanja temelji se na sigurnosnim otocima (engl. *islands of certainty*) i parsanju koje prednost daje jednostavnijim izrazima (engl. *easy-first parsing*). Najprije se označe oni segmenti koji imaju jednostavnu ali istovremeno i sigurnu interpretaciju te se tako kreiraju manji sigurnosni otoci. Prelaskom obrade na sljedeću razinu, povećava se broj označenih sintagmi na osnovi onih prethodno označenih te tako rastu i sigurnosni otoci.

Kako bi se postigla veća učinkovitost parsera, Abney (Abney, 1996a:3) vjeruje da se gramatika ne bi smjela proučavati kao lingvistički opis već kao programski jezik za prepoznavanje. Zato i napominje da je osnovni cilj *‘napisati takve predloške koji pouzdano mogu odrediti djeliće sintaksne strukture čak i ako se radi samo o ‘granici’ ili o ‘srži’ strukture, a ne o tradicionalnim izrazima’*. Kaskadnim modelom, Abney definitivno daje prednost djelomičnom parsanju ali time nikako ne podcjenjuje točnost i brzinu takvoga parsanja.

Kaskadama su se koristili i Buchholz, Veenstra i Daelemans (Buchholz, Veenstra, Daelemans, 1999.) u svom modelu kaskadnog pridruživanja gramatičkih relacija i to na tri načina:

- sljedeća kaskada temelji svoje odluke na informacijama prethodne kaskade;
- prethodna kaskada ograničava broj odluka koje sljedeća treba napraviti;
- prethodna kaskada smanjuje broj elemenata koji se može koristiti u sljedećoj kaskadi.

## 2.5. Višeznačnost

Mnoge riječi imaju više značenja koja ljudi u svakodnevnom govoru filtriraju ovisno o kontekstu u kojem ih čuju ili koriste. Taj se proces odvija spontano i bez velikoga razmišljanja. Mason (Mason, 2004.) čak vjeruje da je govorni jezik rijetko kad višeznačan zahvaljujući dodatnim informacijama koje proizlaze iz zajedničkoga kontekstnog znanja. No, računalo tu nailazi na problem jer se kod njega proces filtriranja, odnosno odabira pravoga značenja, zbiva na drukčiji i duži način uz, mogli bismo reći, dosta ‘razmišljanja’. Mnogi autori vjeruju da se problem višeznačnosti može riješiti tek kad uspijemo riješiti sve teže probleme<sup>83</sup> umjetne inteligencije. Upravo se pronalaženje i rješavanje višeznačnosti smatra jednim od osnovnih izazova sustavima za razumijevanje jezika.

Višeznačnost se pripisuje onim rečenicama koje imaju više od jednog stabla izvođenja, no pri tome razlikujemo dva tipa višeznačnosti:

- bitnu (esencijalnu) višeznačnost,
- lažnu višeznačnost.

Koja je razlika? Kako se već i naslućuje iz njihovih imena, lažno višeznačna je rečenica koja ima više od jednog parsnog stabla, ali svako to stablo opisuje istu temeljnu rečeničnu semantiku. Za razliku od toga, postoje i druge rečenice čija parsna stabla opisuju različite semantike. To su bitne višeznačnosti, odnosno, višeznačnosti koje stvaraju probleme u parsanju.

Prema Jurafskome i Martinu (Jurafsky, Martin, 2000:4) višeznačnost se pojavljuje kad postoji više mogućih lingvističkih opisa za isti izraz. Pogledajmo to na primjeru za engleski jezik:

---

<sup>83</sup> Misli se na prikaz enciklopedijskoga znanja i prikaz općeg znanja (engl. *common sense*).

✎ *I made her duck.*

✎ *može značiti*

✎ *skuhao sam joj patku;*

✎ *napravio sam joj patku;*

✎ *pretvorio sam je u patku;*

✎ *natjerao sam je da se sagne;*

✎ *...*

U ovom primjeru 'duck' je leksički i sintaktički višeznačna riječ (duck – imenica i glagol u značenjima 'patka' i 'sagnuti se'), 'her' je morfološki i sintaktički višeznačna riječ (her – dativ i posvojna zamjenica u značenju 'njoj' i 'njen') dok je riječ 'make' semantički (može imati značenje 'stvoriti', 'pripremiti' i 'natjerati') i sintaktički višeznačna (može biti prijelazni glagol s jednim objektom i dvo-prijelazni glagol s dva objekta).

Leksička se višeznačnost često uspješno rješava unutar sintagme osim ako se radi o sintagmama sa samo jednom riječju, budući da jedna riječ ne daje dovoljno informacija za rješavanje leksičke višeznačnosti.

Jurafsky i Martin navode dva osnovna tipa strukturalne višeznačnosti:

- **vezna višeznačnosti** (engl. *attachment ambiguity*) pojavljuje se kada nije jasno koje se dvije skupine međusobno vežu. Tako npr. rečenica

✎ *I saw an elephant in the park with a telescope.*

- može imati tri parsna stabla od kojih je u jednom skup 'with a telescope' vezan na glavni glagol, u drugom je vezan uz grupu 'an elephant', a u trećem je vezan uz grupu 'the park' pa je interpretacija prvog parsnog stabla da sam ja imao teleskop, a drugog da je slon kojeg sam vidio imao teleskop, a trećeg da je park u kojem sam vidio slona imao teleskop;



- **koordinacijska višeznačnost** (engl. *coordination ambiguity*) se pojavljuje kada kod koordinacijskih riječi (u hrvatskom bi to bili veznici 'i, ili, ni, niti') nije jasno što sve ulazi u koordinaciju. Tako u izrazu

✎ 'stari čovjek i žena'

- nije posve jasno je li samo čovjek star ili je i žena stara.

Za uklanjanje strukturalne višeznačnosti nalazimo kod Karlsoona (Karlsoon, 1995:24) sljedeće metode:

- ručna intervencija;
- statistička optimizacija temeljena na tranzicijskoj vjerojatnosti;
- unifikacija;
- Lambekov račun (engl. *Lambek calculus*);
- primjena neovisnih gramatičkih pravila za usporednu provjeru okolina;
- posebna pravila za uklanjanje višeznačnosti;
- pristup temeljen na bazama znanja;
- itd.

Za rješavanja problema POS višeznačnosti mogu se koristiti i već spomenute lokalne gramatike (vidi poglavlje 2.3.2.1) kojima se može ubrzati POS označivanje ili parsanje teksta.

Ide i Véronis (Ide, Véronis, 1998.) daju pregled metoda za rješavanje višeznačnosti na sljedeći način:

- rane metode korištene za računalno prevođenje (Kaplan 1950, Reifler 1955);
- metode zasnovane na umjetnoj inteligenciji;
- simboličke metode (semantičke mreže, padežni okviri);
- metode zasnovane na znanju;
- računalno čitljivi rječnici (Lesk 1986, Wilks i suradnici 1990);
- tezaursi (npr. Roget's International Thesaurus<sup>84</sup> – 1950tih);

<sup>84</sup> <http://www.bartleby.com/thesauri/>

- 
- računalni leksikoni;
  - numerirani leksikoni (WordNet<sup>85</sup>);
  - generativni leksikoni<sup>86</sup> (CoreLex<sup>87</sup>);
  - konekcijske metode (engl. *connectionist methods*) (model raspršene/prenesene aktivacije<sup>88</sup> (engl. *spreading activation model*));
  - metode zasnovane na jednojezičnom korpusu;
  - uz pomoć paralelnih korpusa;
  - ručno označeni smisao - (Brown korpus - 1967, Trésor de la Langue Française - 1971, Kolhapur korpus<sup>89</sup>, Lancaster-Oslo-Bergen korpus - 1980, British National korpus - 1990);
  - automatski označen smisao;
  - uz pomoć samodopunjujuće metode.

Iako riječi u nekoj rečenici imaju više svojih značenja, njihov međusobni poredak često smanjuje moguće interpretacije. Pojam konteksta ima veliku ulogu u rješavanju višeznačnosti i koristi se na dva načina:

- pristupom «vreća riječi» (engl. *bag of words approach*) – kontekstom se smatraju samo riječi koje s višeznačnom riječi čine grupaciju;
- relacijske informacije (engl. *relational information*) – kontekstom se smatraju relacije s višeznačnom riječi (udaljenost, sintaksne relacije, ortografska svojstva, kolokacije, semantičke kategorije itd.).

Kao pomoć pri odabiru pravog značenja koriste se informacije iz mikro-konteksta ili iz tematskog konteksta. Mikro ili lokalnim kontekstom podrazumijeva se uži broj riječi koje okružuju traženu riječ. Tematski kontekst uključuje samostalne riječi koje se pojavljuju u istom kontekstu kao i

---

<sup>85</sup> <http://wordnet.princeton.edu/perl/webwn>

<sup>86</sup> Za razliku od numeriranih leksikona u kojima je smisao eksplicitno dan, generativni leksikoni koriste pravila za pronalaženje pravilnosti kod kreiranja pojmova kako bi se dobile informacije o smislu traženoga pojma.

<sup>87</sup> <http://www.cs.brandeis.edu/~paulb/CoreLex/corelex.html>

<sup>88</sup> Model raspršene ili prenesene aktivacije aktivira koncepte iz semantičke mreže u trenutku njihove upotrebe nakon čega se aktivacija prenosi na vezne čvorove i prenošenjem slabi.

<sup>89</sup> Korpus za indijski engleski jezik.

tražena riječ unutar par rečenica koje okružuju rečenicu u kojoj se tražena riječ nalazi. Istraživanje koje je proveo Yarowsky 1993. pokazuju da je za različite vrste riječi potrebno koristiti različite kontekste. Tako je na primjer, za imenice dobro koristiti tematski kontekst, no kod glagola i pridjeva, potrebno je gledati riječi što bliže traženoj riječi.

Tip višeznačnosti koji je za potrebe ovog rada interesantniji je POS višeznačnost, za razliku od semantičke višeznačnosti kojom se ovdje neću baviti. Tadić (Tadić, 2003.) razlikuje dvije vrste višeznačnosti, odnosno istopisnosti. Prvu vrstu čini unutarnja istopisnost u koju spadaju one pojavnice koje su u stvari različiti oblici iste leme što je slučaj s mnogim imenicama u hrvatskome jeziku koje imaju iste oblike u dativu, lokativu i instrumentalu množine (npr. imenica 'mora') ili pak u nominativu, akuzativu i vokativu jednine (npr. imenica 'more').

Drugu vrstu čini vanjska istopisnost u kojoj je ista pojava oblik različitih lema. Takva je na primjer imenica MORE koja označava nominativ i akuzativ jednine za pojam koji označava more u kojem se kupamo (Jadransko more). Drugo značenje je ponovo imenica ali u genitivu jednine, nominativu i akuzativu množine pojma koji označava loše snove (noćne more). Ili pak oblik 'MORA' koji označava dativ, lokativ i instrumental množine za imenicu 'mora' (kao 'noćna mora'), genitiv jednine, nominativ, genitiv, akuzativ i vokativ množine za imenicu 'more' (kao 'Jadransko more'), ali i 3.1.j. muški, ženski i srednji rod prezenta i imperativa glagola 'morati'. Sljedeći primjer za vanjsku istopisnost je riječ 'PRIČA' koja može biti imenica u nominativu jednine i genitivu množine, ali i prezent glagola pričati, 3.1.j. muški, ženski i srednji rod. Slično je i s riječi 'DA' koja može biti označena kao veznik i kao prezent glagola dati, 3.1.j. u muškom, ženskom i srednjem rodu.

Istopisnice u postojećem flektivnom podleksikonu za hrvatski jezik raspoređene su na sljedeći način:

Vrsta riječi	Osnovni oblik	Različiti oblici	Različiti likovi
Glagoli	1.150	103.141	24.361

Imenice	19.189	272.243	120.914
Pridjevi	13.863	1.130.923	242.925
Zamjenice	58	2.712	646
Brojevi	68	898	255
Ukupno	33.178	1.509.917	389.101

*Tablica 2 – raspoređenost istopisnica u hrvatskom flektivnom podleksikonu*

Iz tablice se iščitava da u ovom flektivnom leksikonu za hrvatski jezik 76,38% glagola, 55,59% imenica, 78,52% pridjeva, 76,18% zamjenica i 71,60% brojeva ima unutarnju istopisnost ili vanjsku istopisnost unutar iste vrste riječi (kao što je slučaj s riječima MORE i MORA). No, spoje li se različiti oblici za gore navedene vrste riječi, tada je broj različitih likova među njima 385.497 što govori da 0,93% promjenjivih vrsta riječi ima vanjsku istopisnost.

Broj nepromjenjivih vrsta riječi u leksikonu je 433, ali ih je od toga 425 različitoga lika što znači da ih je 1,85% unutar ove klase istopisno, tj. imaju vanjsku istopisnost.

Dodaju li se nepromjenjive vrste riječi promjenjivima, dobit će se da je od ukupno 1.510.350 različitih oblika riječi, njih svega 385.843 različitoga lika, što znači da je u ovome hrvatskome flektivnome leksikonu, kojem su pridružene i riječi bez fleksije, 74,45% istopisnih oblika. Od toga 74,21% odlazi na unutarnju, a 0,24% na vanjsku istopisnost.

U ovoj fazi rada neću obrađivati riječi koje imaju više značenja, ali svako značenje pripada istoj kategoriji riječi, tj. vrsti riječi, kao što je bio primjer s vanjskom istopisnosti za riječ MORE.

No, riječi koje u istom obliku mogu biti različite vrste riječi kao što su primjeri PRIČA i DA pokušat ću razlučiti lokalnim gramatikama. U tome bi od velike pomoći trebale biti okolne riječi, dakle riječi koje prethode i slijede POS višeznačne riječi.

## 2.6. Vrednovanje

Vrednovanje parsera radi se na način da se rezultati koje je dao parser uspoređuju sa zlatnim standardom, tj. s ručno označenim korpusom. Kako je rasla komercijalna vrijednost jezičnih tehnologija, tako je rasla i važnost procjene sustava. Iako je kroz povijest bilo dosta rasprava o načinima mjerenja učinkovitosti sustava, preciznost i odziv isplivali su na površinu kao najčešće korišten par u mjerenju učinkovitosti sustava za obradu prirodnog jezika. Tome je razlog razumljivost rezultata dobivenih na ovaj način.

Jedan od uobičajenih metoda vrednovanja je korištenje PARSEVAL<sup>90</sup> mjera u koje ulaze preciznost, odziv i nepravilno zatvorene zagrade<sup>91</sup> (engl. *Cross Brackets*). Ja ću za potrebe ovog rada pak koristiti, uz preciznost i odziv, f-mjeru umjesto nepravilno zatvorenih zagrada.

Preciznost i odziv po prvi je put definirao Cyril Cleverdone krajem 1950-ih za potrebe evaluacije sustava za pronalaženje obavijesti dok je f-mjeru definirao van Rijsberg 1979. godine uzimajući u obzir važnost odziva naspram preciznosti.

Dobiveni rezultati prikazuju pripadnost pojedinih rečeničnih sastavnica odgovarajućim skupinama (NP, VP, PP), ali ne i ovisnosne veze između sastavnica. Tako je vrijednost za odziv 100% ako su sve riječi morfološki i sintaksno ispravno označene, a preciznost ima vrijednost 100% ako u izlazu ne postoje morfološki ili sintaksno neispravno označene riječi.

Postojanje PARSEVAL mjernog sustava omogućava uspoređivanje parsera koji koriste različite gramatike, različite algoritme, odnosno, primjenjeni su na različitim jezicima.

---

<sup>90</sup> Naziv PARSEVAL dolazi od engleskog izraza PARSer EVALuation.

<sup>91</sup> Metodom nepravilno zatvorenih zagrada mjeri se prosjek rečenica u kojima ne dolazi do presjecanja zagrada kojima se označavaju rečenični skupovi.

### 2.6.1. Preciznost

Van Rijsberg (van Rijsberg, 1979.) preciznost računa kao omjer ukupnog broja pronađenih podataka koji su relevantni i ukupnog broja pronađenih podataka:

$$\text{preciznost} = \frac{\text{Relevantni i pronađeni}}{\text{pronađeni}}$$

Mikheev (Mikheev, 2003:203) definira preciznost kao omjer točnih odgovora i ukupno dobivenih odgovora dok je Hirschman i Mani (2003:416) definiraju kao broj točno detektiranih instanica unutar ukupnog broja detektiranih instanica.

Preciznost je definirana i u (Jurafsky, Martin, 2008:455) gdje nalazimo po dvije definicije kako za preciznost tako i za odziv. Prva se odnosi na vrednovanje parsera, a druga na vrednovanje označenih sintagmi.

Za izračun preciznosti parsera, potrebni su nam podaci iz 'zlatnog standarda', odnosno:

$$\text{Preciznost} = \frac{\text{broj točnih sastavnica ispitnog teksta}}{\text{ukupan broj sastavnica ispitnog teksta}}$$

gdje je sastavnica ispitnog teksta označena kao točna ako u zlatnom standardu postoji ista takva (što znači da ima isti početak i kraj).

Mjerenjem preciznosti označenih sintagmi dobije se postotak računalno pronađenih točnih sintagmi. Sintagma je točno pronađena ako su joj točno određene granice i ako joj je točno pridružena oznaka skupa.

$$\text{Preciznost} = \frac{\text{broj točno pronađenih sintagmi}}{\text{ukupan broj sintagmi koje su pronađene}}$$

U svakom slučaju, bez obzira na to koji tip podatka gledamo, ako je rezultat za preciznost manji od 100%, znači da je određeni broj elemenata ostao netočno označen u tekstu.

### 2.6.2. Odziv

Odziv je prema Cleverdonu (van Rijsberg, 1979:111) omjer relevantnih podataka naprema ukupnom broju pronađenih podataka. Van Rijsberg to prikazuje kao:

$$\text{odziv} = \frac{\text{relevantni i pronađeni}}{\text{ukupan broj relevantnih}}$$

Mikheev (Mikheev, 2003:203) definira odziv kao omjer točnih odgovora i ukupno očekivanih odgovora, a Hirschman i Mani (2003:416) ga definiraju kao broj točno detektiranih instanca unutar ukupnog broja instanca koji se trebao detektirati.

Definicija odziva za parsere prema Jurafsky i Martinu izgleda ovako:

$$\text{Odziv} = \frac{\text{broj točnih sastavnica ispitnog teksta}}{\text{broj točnih sastavnica u zlatnom standardu}}$$

dok definicija odziva za označene sintagme izgleda ovako:

$$\text{Odziv} = \frac{\text{broj točno pronađenih sintagmi}}{\text{stvaran broj sintagmi u tekstu}}$$

To znači da ako je rezultat za odziv manji od 100% određeni broj elemenata u tekstu je ostao neoznačen.

### 2.6.3. F-mjera

Povezivanjem odziva i preciznosti dobila se nova mjera koju van Rijsbergen naziva f-mjerom, a definira je na sljedeći način:

$$F = \frac{(\beta^2 + 1) (\text{preciznost} * \text{odziv})}{\beta^2 (\text{preciznost} + \text{odziv})}$$

Parametar  $\beta$  označava važnost odziva i preciznosti. U našim mjerenjima njihova je važnost podjednaka pa je vrijednost za  $\beta = 1$ . Stoga formulu možemo napisati kao:

$$F = \frac{2 (\text{preciznost} * \text{odziv})}{(\text{preciznost} + \text{odziv})}$$

Izračunom f-mjere računamo uspješnost sustava za obradu jezika odnosno predstavljamo harmonijsku sredinu preciznosti i odziva kao što je opisano u (Bekavac, 2005.) i (Agić, Tadić, 2006.).



### **3. PRAKTIČNI DIO MODELA**

### 3.1. Izvedba

Za izradu prvoga prijedloga modela parsera za hrvatski jezik odlučila sam koristiti NooJ<sup>92</sup> razvojno okruženje kao logičan nastavak na disertaciju Bože Bekavca (Bekavac, 2005.). LFG model za rečeničnu analizu prikazanog u (Seljan, 2003.) i sustav OZANA kojeg je Bekavac prikazao u svom doktorskom radu možemo smjestiti na početke računalne sintaksne analize hrvatskoga jezika.

Moj je cilj bio proširiti OZANA sustav s prepoznavanjem ostalih sintagmi u hrvatskome jeziku uključujući imenske, glagolske, pridjevske i prijedložne sintagme te prikazati barem neke od njihovih međusobnih odnosa. Prije svega tu mislim na odnos između subjekta i predikata u rečenici, te predikata i direktno i/ili indirektnog objekta. U tome mi je pomogao i hrvatski glagolski valencijski sustav, CROVALLEX 2.0008, kojeg je izradila i u svojoj disertaciji prikazala Nives Mikelić Preradović (Mikelić Preradović, 2008).

Od Mikelić Preradović preuzeti su valencijski opisi 1739 hrvatskih najfrekventnijih glagola koje je ona preuzela iz hrvatskoga frekvencijskoga rječnika (Moguš, Bratanić, Tadić, 1999.). Taj sam popis dodatno proširila glagolima koje sam već imala u postojećem leksikonu, ali se do sada nisu nalazili u CROVALLEX-u.

Leksikon sam preuzela iz *Hrvatskog morfološkog leksikona* čije je temelje dao Tadić u (Tadić, 1994., 2003. i 2005.) i prilagodila ga NooJ notaciji. Svakoj pojavnici u leksikonu trebalo je odrediti flektivni model koji je potom trebalo i opisati (vidi Prilog 3 za opis imenskih paradigmi, Prilog 4 za opis pridjevskih paradigmi, Prilog 5 za opis glagolskih paradigmi, Prilog 6 za opis paradigmi za brojeve i zamjenice).

Za opis sintagmi u hrvatskome jeziku koristila sam lokalne gramatike. One se kaskadno povezuju u konačan parser. Njihov kaskadni poredak je bitan jer se njime najprije pokušava lokalno riješiti višeznačnost na razini

---

<sup>92</sup> NooJ aplikacija je novija verzija Intex aplikacije koju je u svom radu koristio i B. Bekavac (2005).

označivača vrsta riječi, potom na razini sintagmi, te konačno na najvišoj sintaksoj razini. Grafički bi se to moglo prikazati na sljedeći način u kojem izlaz iz prethodne razine predstavlja ulaz za sljedeću razinu obrade:

opojavničenje

↳ određivanje sintagmi

↳ parsanje.

### 3.1.1. Korpus za testiranje

Korpus za testiranje preuzela sam iz 100 Kw hrvatskog dijela hrvatsko-engleskog paralelnog korpusa (Tadić, 2000b) kojeg čine novinski članci objavljeni u časopisu *Croatia Weekly* od početka 1998. do 2000. godine. Teme koje časopis obrađuje su dosta široke i pokrivaju područja poput politike, ekonomije, turizma, kulture i sporta.

#### 3.1.1.1. Zlatni standard za razdjelnik

Uzorak na kojem smo testirali razdjelnik sastoji se od 135 nasumično odabranih rečenica (vidi Prilog 1). Kao što se vidi u tablici koja slijedi, u tekstu je 3075 pojava od kojih je najfrekventnija pojava 'je' s čak 133 pojavljivanja iza koje slijede pojavnice 'i', 'u', 'da' itd. U tekstu se nalazi i 224 ponavljajuća dvograma od kojih su najfrekventniji parovi 'da se', 'da će' i 'da je'. U tablici dajem pregled deset najfrekventnijih pojava i dvograma unutar uzorka teksta koji predstavlja zlatni standard.

3075 pojava		224 dvograma koji se ponavljaju	
Frekvencija	Pojavnica	Frekvencija	Dvogram
133	je	11	da se
93	i	11	da će
78	u	10	da je
58	da	8	u Hrvatskoj
47	se	7	će se
46	za	7	je da
41	na	6	što je

38	a	6	bi se
29	će	6	milijuna kuna
26	su	5	koji je

**Tablica 3**

10 najfrekventnijih pojava i dvograma u zlatnom standardu za razdjelnik

### 3.1.1.2. Zlatni standard za parser

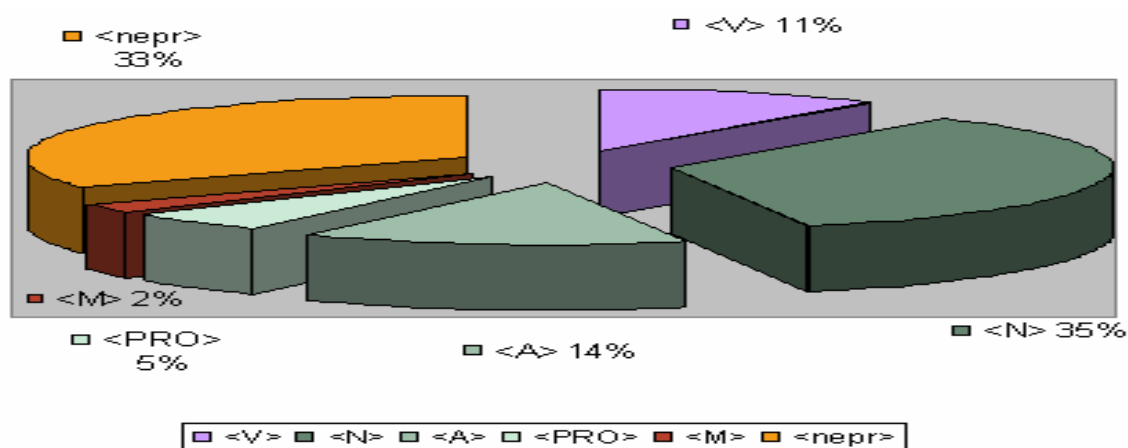
Uzorak na kojem sam testirala parser sastoji se od 146 ručno odabranih jednostavnih rečenica koje sam pronašla u gore opisanom korpusu (vidi Prilog 2). U ovom se uzorku nalazi 1460 pojava od kojih je najfrekventnija 'u', te od ukupno 1415 dvograma, 34 su ponavljajuća dvograma od kojih je najfrekventniji 'u Hrvatskoj'. Slijedi tablica s 10 najfrekventnijih pojava i dvograma.

1460 pojava		34 dvograma koji se ponavljaju	
Frekvencija	Pojavica	Frekvencija	Dvogram
58	u	5	u Hrvatskoj
43	i	4	to je
42	je	3	u svojoj
28	za	3	odnosu na
20	se	3	se u
16	na	3	Haaškim sudom
15	su	2	stranke i
15	s	2	Jedan od
13	od	2	pred Haaškim
11	o	2	su se

**Tablica 4**

10 najfrekventnijih pojava i dvograma u zlatnom standardu za parser

U tekstu nije bilo za parser nepoznatih riječi. Vrste riječi su u tekstu raspoređene na sljedeći način: 35% riječi su imenice <N>, 14% pridjevi <A>, 11% glagoli <V>, 5% zamjenice <PRO>, 2% brojevi <M> dok preostalih 33% čine nepromjenjive vrste riječi <nepr> (prijedlozi, prilozi, veznici i čestice). U tekstu (sasvim slučajno) nema uzvika kao vrsta riječi. Međusobni odnos vrsta riječi u tekstu prikazan je grafom (Slika 14).



Slika 14: Odnos vrsta riječi u zlatnom standardu za parser

Ovaj uzorak predstavlja zlatni standard za testiranje parsera. Rečenice iz zlatnog standarda su i ručno označene kako bi bilo moguće usporediti i vrednovati učinak parsera. Ti su rezultati prikazani u poglavlju 3.5.

## 3.2. Alati korišteni za izradu modela parsera za hrvatski jezik

U izradi modela parsera za hrvatski jezik koristila sam tri alata: NooJ, PrepareForNooj i XMLtoDIC. Autor prvog alata je prof. Max Silberztein, autor drugog alata je Kristina Vučković, a koautori trećeg alata su Zdravko Dovedan i Kristina Vučković.

Izrada posljednja dva alata bila je potrebna kako bi se već postojeći resursi za hrvatski jezik mogli prilagoditi NooJ notaciji.

### 3.2.1. NooJ

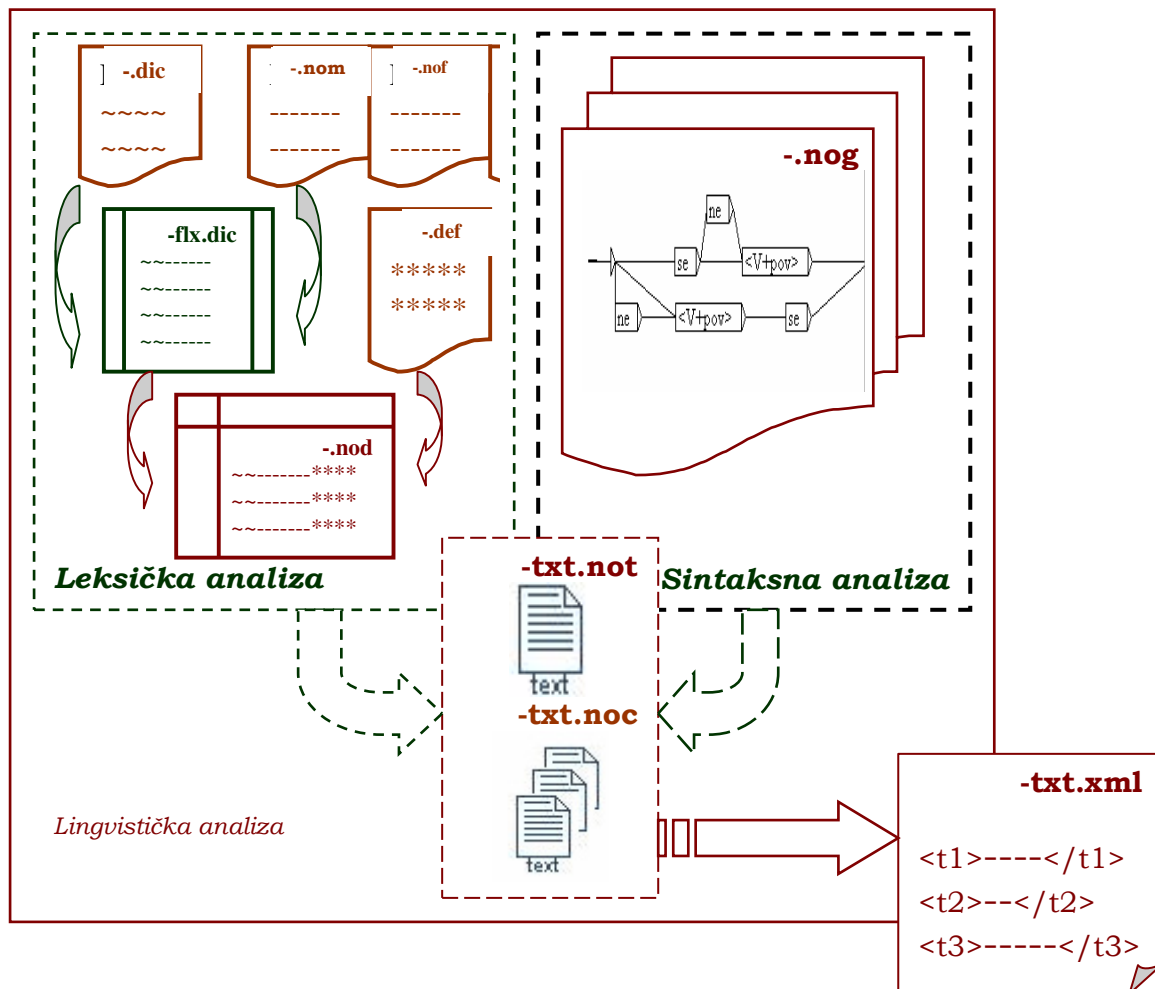
NooJ je razvojno okruženje namijenjeno izradi formalizama za opis prirodnih jezika i njihovoj primjeni na korpusu.

NooJ koristi morfološke i sintaksne **pretvarače konačnih stanja** za dobivanje lingvističkih i leksičkih informacija, **automate konačnih stanja** za postavljanje upita o morfo-sintaksnim pojavljivanjima u korpusu, **regularne izraze** za jednostavne upite nad korpusom (konjukcija, disjunkcija, negacija, Kleenova zvijezda), **bezokolinske gramatike** za alternativni pristup izradi gramatika (vidi primjer \*.nof datoteke) i **rekurzivne mreže prijelaza** (gramatike s više od jednog grafa konačnih stanja – bilo da se radi o automatu ili pretvaraču), **unaprijeđene rekurzivne mreže prijelaza** (engl. *Enhanced Recursive Transition Networks* – ERTN) čije korištenje varijabli omogućava provođenje različitih operacija nad izlazom.

Osnovni jezični izvori u NooJ-u su:

- **rječnik** (\*.dic) koji, uz popis osnovnih oblika riječi, sadrži i osnovne informacije o njima (kategoriju riječi, flektivnu paradigmu, sintaksna i semantička svojstva) i

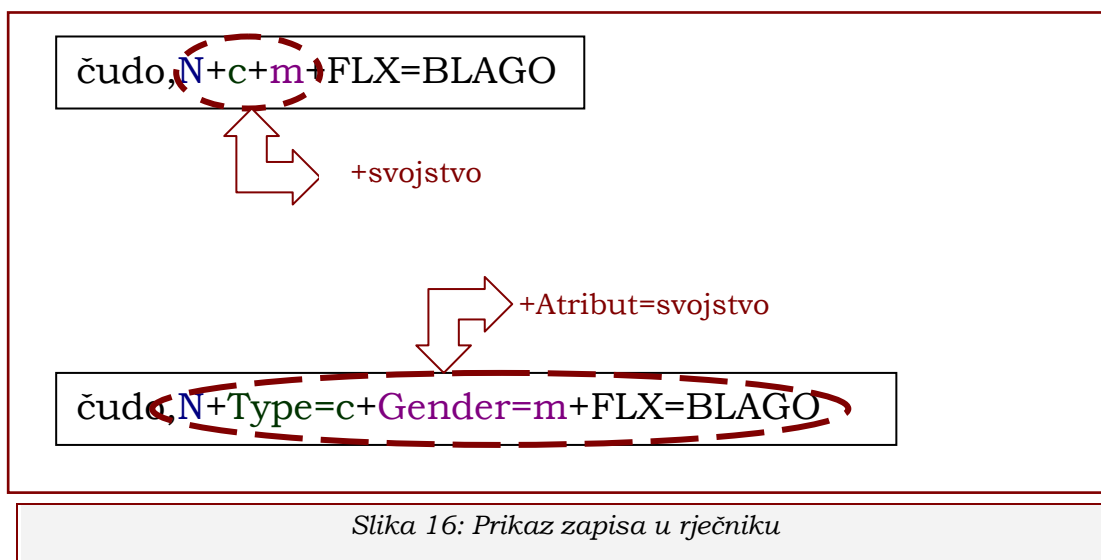
- **gramatika** za opis različitih lingvističkih fenomena koja dolazi u tri različita oblika:
- **morfološke gramatike** (\*.nof) u obliku grafa ili pravila opisuju flektivne paradigme (deklinacije, konjugacije i komparacije) i paradigme izvođenja;
  - **leksičke, ortografske i terminološke gramatike** (\*.nom) povezuju oblike riječi s njihovim leksičkim informacijama;
  - **sintaksne i semantičke gramatike** (\*.nog) označavaju izraze u tekstu, a koriste se i za uklanjanje višeznačnosti.



Slika 15: Lingvistička analiza u NooJ-u

Svaki od navedenih jezičnih izvora ima i zasebnu notaciju čiji opis je detaljno prikazan u poglavlju 3.3.

U rječniku se svaka riječ navodi u zasebnom retku. Osnovni oblik riječi odvaja se zarezom od svog opisa. Svojstva riječi dodaju se uz pomoć '+' znaka i mogu se navoditi u kraćem formatu ('+svojstvo') ili dužem formatu ('+atribut=svojstvo') kao što je prikazano na Slici 14.



Slika 16: Prikaz zapisa u rječniku

U flektivnoj gramatici moguće je paradigme opisivati uz pomoć grafa ili linearnog zapisa. Ja sam se odlučila za ovaj drugi model.

Za paradigmu je potrebno navesti njezino ime iza kojeg slijedi znak jednakosti '=', a potom opis paradigme. Svaka paradigma na svome kraju ima znak ';'. Opis paradigme sastoji se od dva dijela međusobno odvojena znakom '/'. Ispred kose crte navodi se promjena koja se treba dogoditi osnovnom obliku riječi koji se nalazi u rječniku, a iza kose crte dolazi opis generirane riječi. Ovisno o vrsti riječi koju opisuje, opis uključuje padež i broj za imenice, padež, broj, rod i komparaciju za pridjeve, padež, broj i rod za brojeve i zamjenice, te broj, rod, lice i glagolsko vrijeme za glagole. Tako npr. opis jednine paradigme imenice BLAGO glasi:

*BLAGO* = <E>/Nom+s + <B1>(a/G+s + u/D+s) + <E>/Acc+s  
+ <E>/Voc+s + <B1>(u/L+s + om/I+s);

odnosno,



- nominativ jednine tvorimo na način da se osnovni oblik ne mijenja i da na takav oblik riječi dodamo opis kojim se obilježava nominativ (Nom) jednine (s),
- genitiv jednine tvorimo na način da se izbriše jedno zadnje (<B1>) slovo osnovnog oblika a potom doda nastavak 'a',
- dativ jednine tvorimo na način da se izbriše jedno zadnje slovo osnovnog oblika a potom doda nastavak 'u',
- akuzativ jednine se tvori isto kao i nominativ jednine,
- vokativ jednine se tvori isto kao i nominativ jednine,
- itd.

Oznaka <B1> koja se nalazi ispred male zagrade za genitiv i dativ jednine, kao i za lokativ i instrumental jednine, znači da se uputa o brisanju jednog zadnje slova odnosi na sve opise koji se nalazi unutar zagrada. Osim oznake <Bx><sup>93</sup> i oznake <E> koristila sam još i

- <LW> za odlazak na početak riječi (kod npr. kreiranja superlativa pridjeva gdje se, osim na kraj riječi, treba još dodati i prefiks 'naj'),
- <Sx> za brisanje x trenutnih znakova i
- <Rx> za pomicanje u desno x znakova.

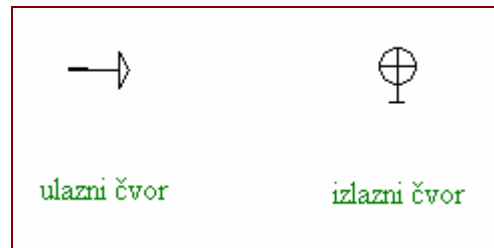
Primjer za korištenje <Rx> i <Sx> je u kreiranju komparativa paradigme BIJEL koji glasi:

✎  $BIJEL = ji<LW><R1><S1>/Nom+m+s+cp + \dots;$

a koji nas nakon dodavanja 'ji' na kraj osnovnog oblika BIJEL šalje na početak riječi <LW>, zatim se pomiče za jedno mjesto u desno <R1> (u našem primjeru, smješta se između 'B' i 'T') te briše jedan znak koji slijedi <S1> (u ovom je slučaju to znak 'i') kako bi smo na kraju dobili 'bjelji' kao oblik za komparativ pridjeva u nominativu muškog roda u jednini.

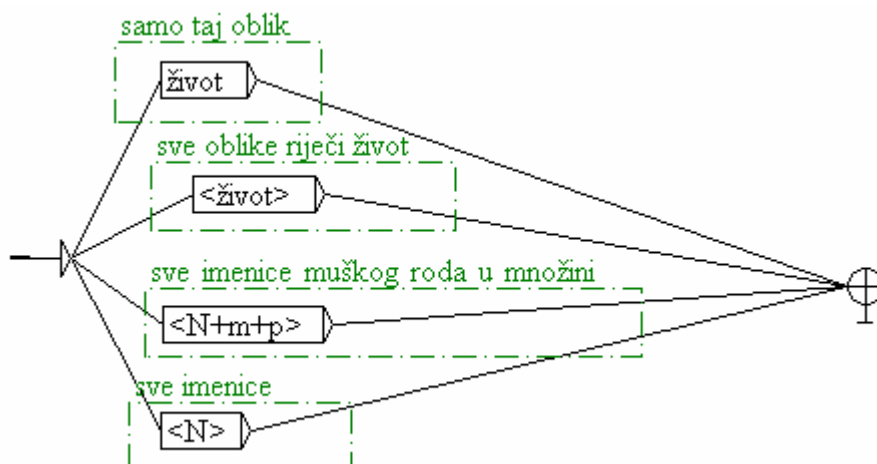
Za opis lokalnih gramatika koristili smo se grafičkim zapisima. Svaki graf ima jedan ulazni čvor i jedan izlazni čvor (Slika 17).

<sup>93</sup> X označava broj znakova.



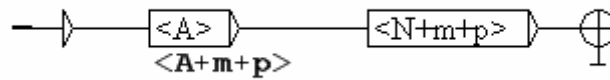
Slika 17: Ulazni i izlazni čvor

Čvorovi koji se nalaze između ova dva obavezna čvora mogu biti jednostavni ili složeni. Jednostavni čvorovi mogu sadržavati oblik jedne riječi, sve oblike jedne riječi, čitavu klasu riječi, čitavu podklasu neke klase (Slika 16) ili negaciju svega navedenoga. Negaciju dobijemo koristeći znak '!' ispred onoga što se negira pa bi u slučaju da naša gramatika sa Slike 18 prihvata sve riječi koje nisu glagol, unutar čvora napisali !<V>.



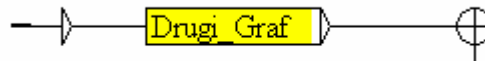
Slika 18: Sadržaj čvorova

Svaki čvor može imati svoj izlaz (ako se radi o prikazu pretvarača) koji se nalazi ispod oznake čvora. Tako u našem primjeru (Slika 19) svaki atribut koji se nađe ispred imenice u muškom rodu u množini treba biti označen kao atribut u muškom rodu u množini.



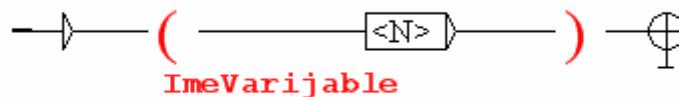
Slika 19: Prikaz pretvarača

Složeni čvor predstavlja vezu na drugi graf tj. svoj podgraf. To je u grafu naznačeno žutom bojom čvora kao što je vidljivo i na Slici 20.



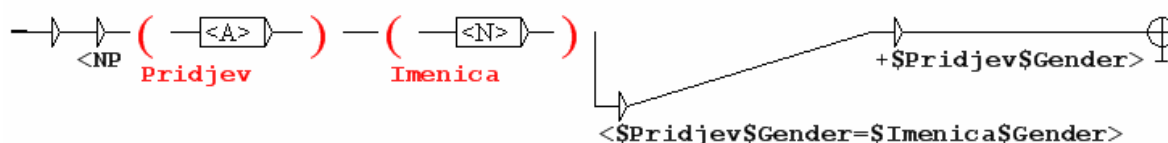
Slika 20: Sadržaj čvora je drugi graf

Vrijednost jednog ili više čvorova može se staviti u varijablu (Slika 21) s kojom možemo provjeravati vrijednosti svakog pojedinog atributa tog čvora (Slika 22). Varijable se označavaju otvorenom i zatvorenom malom zagradom u crvenoj boji dok se provjere vrijednosti atributa ispisuju van čvora.



Slika 21: Prikaz varijable

Na Slici 22 imamo primjer u kojem je vrijednost čvora **<A>** stavljena u varijablu **Pridjev** dok je vrijednost čvora **<N>** u varijabli **Imenica**. U sljedećem koraku **<\$Pridjev\$Gender=\$Imenica\$Gender>** provjeravamo imaju li te dvije varijable istu vrijednost za atribut rod (\$Gender). U slučaju da je rezultat tog upita istinit, tada se cijeli izraz (**<A><N>**) tj. pridjev i imenica koja mu slijedi i s kojom se slaže u rodu, označavaju kao imenska sintagma **<NP>** kojoj pridružujemo vrijednost atributu rod pa je njezina konačna oznaka **<NP+m>**, **<NP+f>** ili **<NP+n>** ovisno o rodu pridjeva.



Slika 22: Provjera vrijednosti atributa varijable

Trenutno se uz hrvatski modul radi na razvijanju i usavršavanju modula za francuski, engleski, talijanski, mađarski, bugarski, makedonski, portugalski, španjolski, srpski, pa čak i za arapski, latinski, armenski i kineski.

### 3.2.2. PrepareForNooj

Za hrvatski jezik već postoji flektivni leksikon (Tadić, 2003., 2005.) koji je označen prema MULTEXT\_East standardu (Erjavec, 2004). Uz pomoć PrepareForNooJ alata, postojeći hrvatski flektivni leksikon konvertiran je u NooJ u dva koraka.

U prvom je koraku trebalo iz flektivnoga leksikona u kojem se nalaze svi oblici hrvatskih promjenljivih riječi (svi padeži imenica, sva vremena glagola, svi oblici pridjeva) ekstrahirati osnovne oblike riječi, izdvojiti zajednički korijen te odvojiti nastavke zajedno s opisom fleksije koju tvore, npr.

✎ želja, N+c+f  
✎ \* želj  
✎ \* a/Nom+s + e/G+s + i/D+s + u/Acc+s + o/Voc+s + i/L+s +  
om/I+s + e/Nom+p + a/G+p + ama/D+p + e/Acc+p +  
e/Voc+p + ama/L+p + ama/I+p;

Kao što se i vidi u navedenom primjeru, za imenicu želja kreiran je osnovni oblik i opis u kojem su kratice definirane na sljedeći način: N – imenica, c – opća (common), f – ženski rod (feminine). Zatim je izveden nepromjenjivi dio riječi koji je uvijek fiksiran u odnosu na nastavke koji slijede s opisom padeža i oznake za jedninu i množinu (detaljniji opis svake kratice

The screenshot shows the 'PrepareForNooJ' program interface. It features a table of word forms for the noun 'želja' and a list of paradigms.

Id	Lena	Oblik	Kategorija	Tip	Rod	Broj	Padež	Konsonant	Klitika	Znak	ZaPadež
3414	žela	žela	N	c	m	1	3	s	-	-	-
3415	želi	želi	N	c	m	2	1	s	m	p	-
3416	želi	želi	N	c	f	2	1	s	n	p	-
3417	želi	želi	N	c	f	2	2	s	n	p	-
3418	želi	želi	N	c	m	2	2	s	n	p	-

Id	Lena	Oblik	Kategorija	Tip	Rod	Broj	Padež	Konsonant	Klitika	Znak	ZaPadež
3528	žuevija	žuevija	N	c	n	1	1	s	-	-	-
3707	žuevija	žuevija	N	c	n	2	1	s	-	-	-
3752	žuevija	žuevija	N	c	n	2	2	s	-	-	-
3768	žuevija	žuevija	N	c	n	2	3	s	-	-	-

**PARADIGME ...**

R	Članovi	Lena	Nastavci
59	ČUVST	čuvstvo	= <B2>ava/g/p + <B2>va/s/p + <B2>v
60	ČUŠLJAV	čušljavac	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
61	ČUČURIN	čučin	= <E>/h/s + <E>u/g/p + <E>/a/s + <E>
62	ČUČ	čučin	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
63	ČUČANJ	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h
64	ČUČAV	čučin	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
65	ČUČENJ	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h
66	ČUČKANJ	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h
67	ČUČAR	čučin	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
68	ČUČARENJ	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h
69	ČUČOK	čučin	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
70	ČUČOR	čučin	= <E>/h/s + <E>/h/s + <E>u/g/p + <E>
71	ČUČOR	čučin	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
72	ČUČORAVO	čučin	= <B2>ac/h/s + <B2>ac/g/p + <B2>ci
73	ČUČORIN	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h
74	ČUČORIŠT	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h
75	ČUČENJ	čučin	= <B1>u/s/p + <B1>u/g/p + <B1>u/h

Slika 23: radno sučelje u 'PrepareForNooJ' programu

dan je u poglavlju 3.3.1.).

Nakon što je za svaku riječ opisana paradigma (vidi Sliku 23), u popisu paradigmi se našlo dosta ponovljenih opisa. Zato je u drugom koraku trebalo usporediti sve izvedene nastavke unutar pojedinačne kategorije riječi i odvojiti po samo jedno pojavljivanje istog opisa (vidi Sliku 24). Taj se opis zadržao kao predstavnik svoje flektivne kategorije i dodao svim riječima u osnovnom leksikonu kao tip paradigme. Tako se na već postojeći zapis, u navedenom primjeru za imenicu 'želja', dodala oznaka +FLX=ABECEDA kao naznaka da ta imenica preuzima sve flektivne nastavke koje ima i riječ

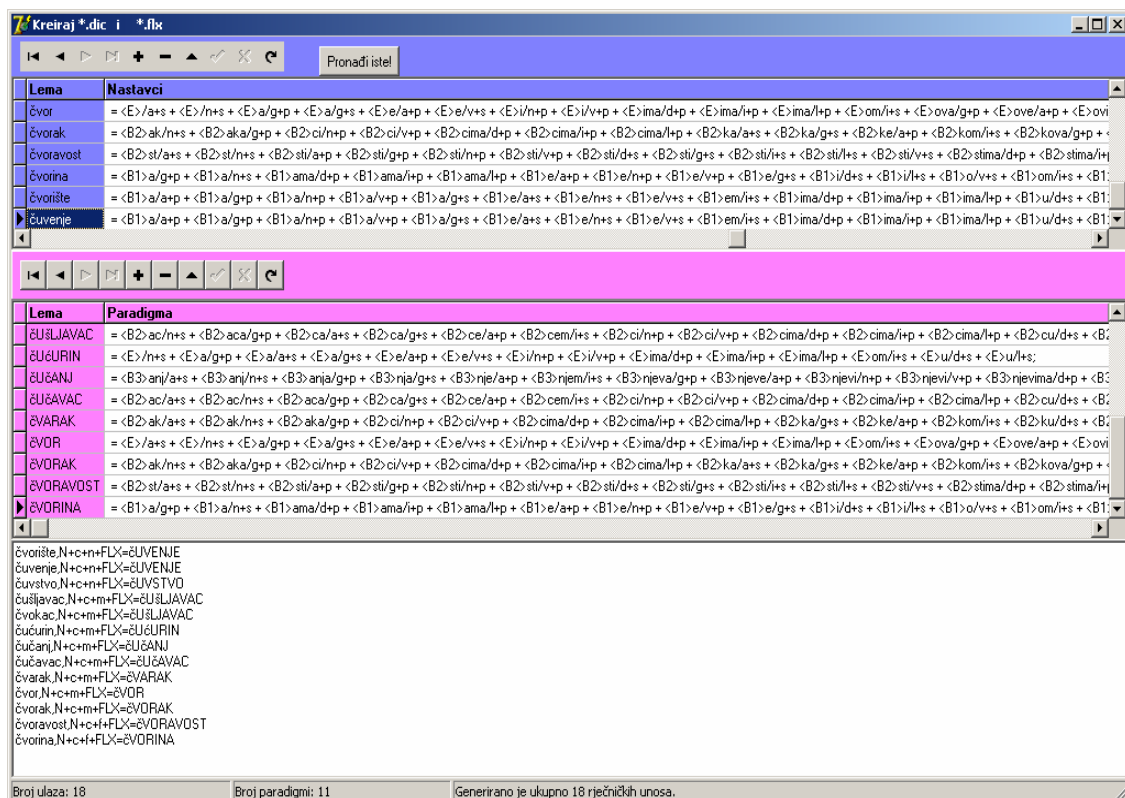
ABECEDA i na isti način kao i riječ ABECEDA. Konačan zapis sada izgleda ovako:

➤ želja, N+c+f+FLX=ABECEDA

što je u skladu s NooJ notacijom.

Kao konačan izlaz iz NooJPrepare programa, dobiju se dvije NooJ datoteke: \*.dic i \*.nof. U prvoj se nalazi hrvatski leksikon s popisom riječi, oznakom njihove kategorije, potkategorije i paradigme, te oznake roda za imenice. Na žalost, nije bilo moguće zadržati originalne MULTEXT-East oznake u potpunosti zbog udvajanja oznaka koje NooJ ne podržava. Stoga je bilo potrebno preimenovati određene oznake. No, jednoznačana obosmjerna konverzija je moguća. Potpun popis s opisom dan je u poglavlju 3.3.3.

Druga datoteka sadrži popis i opis svih paradigmi ovisno o uzorku promjene riječi koju opisuju.

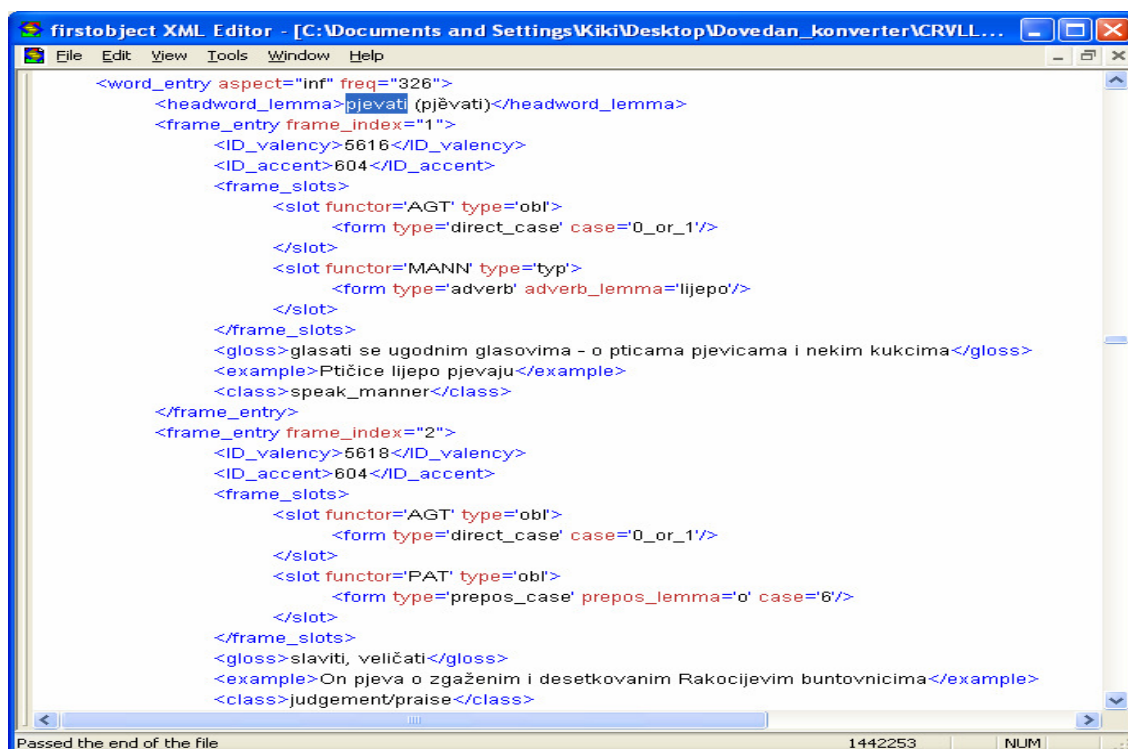


Slika 24: uspoređivanje nastavaka i dodavanje paradigme (FLX=x)



### 3.2.3. XMLtoDIC

Početkom 2008. godine, završen je rad na prvoj fazi izrade hrvatskoga valencijskoga leksikona - CROVALLEX (Mikelić Preradović, 2008.) kada su i podaci iz njega postali dostupni. U računalnom modelu leksikona, trenutno se nalazi 1739 najučestalijih hrvatskih glagola<sup>94</sup> i opis njihovih valencijskih okvira. Svaki okvir sadrži opis površinski realiziranih dopuna koje mogu biti obavezne ili izborne (tipične) i mogu se realizirati kroz padež, prijedlog+padež, prijedlog+padež+infinitiv, zavisnu rečenicu, infinitiv, infinitiv+padež, pridjevsku konstrukciju i konačno kroz priložnu konstrukciju. No, uz ove podatke, svaki okvir sadrži i podatke poput oznaka dubinskih padeža, primjer iz korpusa, idiome itd. (vidi Sliku 25).



Slika 25: XML zapis CROVALLEX-a

Svi podaci mogu se generirati kao xml datoteka. S obzirom da mi nisu bili potrebni svi podaci koji se nalaze u CROVALLEX izlaznoj xml datoteci,

<sup>94</sup> Popis je rađen prema Hrvatskom čestotnom rječniku (Moguš, Bratanić, Tadić, 1999.) u kojem je ukupno 9500 glagola, no njih 1739 ima frekvenciju veću od 11, što je i bio kriterij autorice za odabir glagola u ovoj prvoj fazi izrade leksikona..

bilo je potrebno kreirati program koji će iz te datoteke izvući samo potrebne podatke koje je potom trebalo ispisati u NooJ čitljivom formatu.


Iz tog je razloga kreiran XMLtoDIC alat. Njegov osnovni zadatak bio je konvertiranje xml zapisa iz CROVALLEX-a u NooJ notaciju, posebice za datoteku \*.dic. S obzirom da je leksikon za hrvatski jezik već bio kreiran, bilo je potrebno proći kroz postojeću \*.dic datoteku i postojeće zapise glagola proširiti samo unaprijed dogovorenim oznakama iz CROVALLEX-a. Između postojećih CROVALLEX informacija mene su za ovu fazu obrade jezika interesirali samo sljedeći podaci:

- Je li glagol povratan ili ne
- Veže li na sebe prijedložni skup
  - u kojem padežu
  - obavezno ili izborna vezanje
- Veže li na sebe neprijedložni skup
  - u kojem padežu
  - obavezno ili izborna vezanje.

Konverzija podataka rađena je u dva koraka. Najprije su iz XML zapisa izvučeni samo potrebni opisi: glavni glagol, povratnost glagola, obavezna padežna veza s drugom vrstom riječi, neobavezna padežna veza s drugom vrstom riječi. U drugom koraku su se uspoređivali već postojeći popis glagola u NooJ-u i novonastali popis i opis glagola iz CROVALLEX-a. Kod zajedničkih ulaza osnovnih oblika glagola, opis glagola u NooJ-u bi se proširio s opisom iz CROVALLEXA na sljedeći način. Za primjer glagola SLIČITI, na već postojeći zapis u NooJ-u:

 *sličiti*, V+FLX=ŽAGORITI

dodaje se opis glagola iz CROVALLEX-a

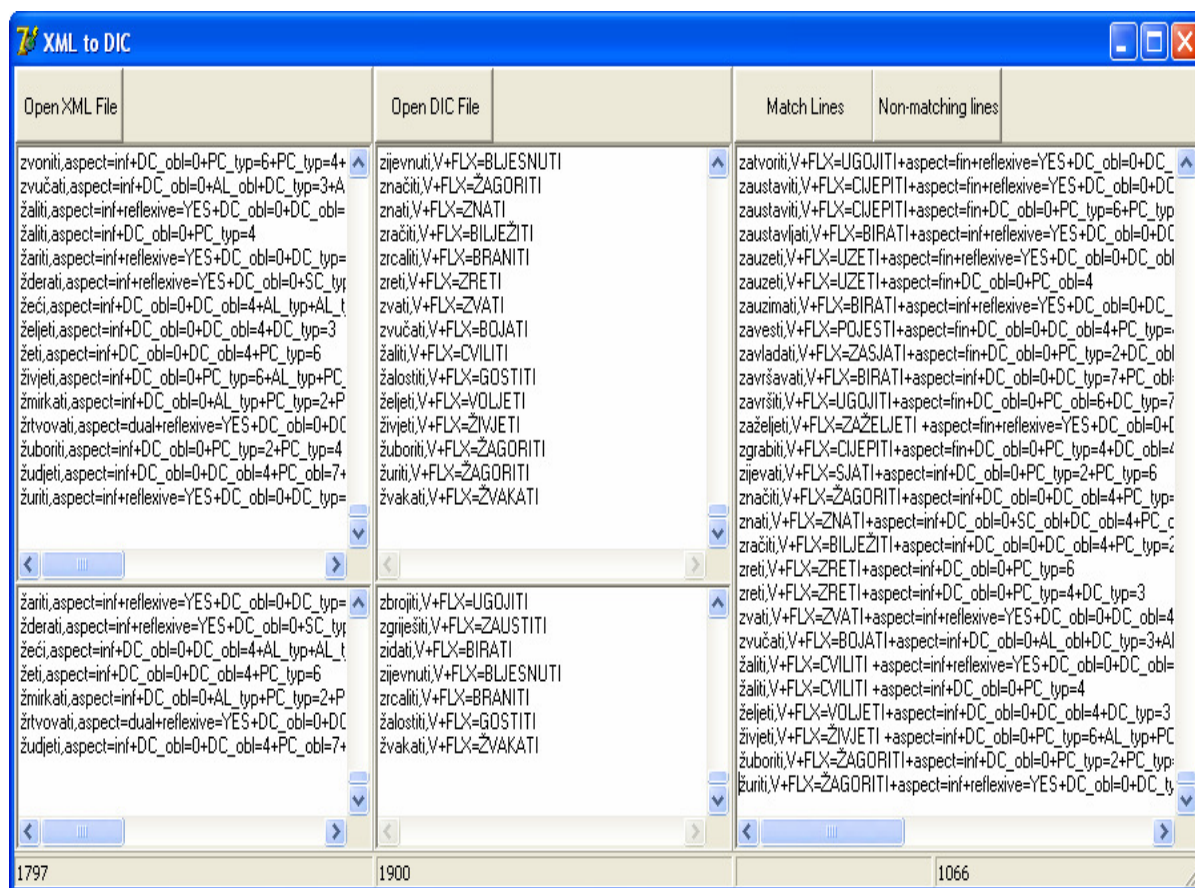
 *sličiti*, Aspect=inf+DCobl=0+DCobl=D

kako bi se dobio zapis

 *sličiti*, V+FLX=ŽAGORITI +Aspect=inf+DCobl=0+DCobl=D



Iz ovog zapisa se iščitava da se glagol (V) '**sličiti**' mijenja prema modelu fleksije glagola ŽAGORITI, da ima infinitivni aspekt (radi se dakle o nesvršenom glagolu), da ima obavezan besprijedložni dodatak u dativu i da ima obavezan besprijedložni dodatak u nominativu koji može biti neizrečen.



Slika 26: Konverzija CROVALLEX XML zapisa u NooJ \*.dic zapis

Nakon konverzije, ustanovila sam da se od 1900 glagola koji se već nalaze u leksikonu njih 1066 nalazi i u CROVALLEXU te je njima jednostavno pridružen dodatni opis (vidi Sliku 26).

S obzirom da CROVALLEX u ovoj fazi izrade sadrži samo glagole koji su prema rječniku iz 1999. godine bili najfrekventniji, 834 glagola iz postojećeg popisa u prvom prolazu nije bio dodatno označeno. Kako se ipak radi o velikom broju glagola, za potrebe izrade ovog doktorata ručno sam dodala opise samo onih glagola koji su se nalazili u testnom korpusu.

Vjerujem da će se opisi ostalih hrvatskih glagola vremenom dodati kako se bude nastavilo s radom na razvoju oba modela.

### 3.3. Leksička analiza

#### 3.3.1. Leksikon

Parser kojeg ovdje opisujem je leksikaliziran što znači da je vezan uz leksikon iz kojeg se izvlače dodatne informacije o svakoj riječi koja se pojavi u tekstu. Leksikon, odnosno rječnik, sadrži 35582 leme, od toga:

Vrsta riječi	Kratika	Broj lema	Broj oblika
Imenica	N	19 681	274 300
Glagol	V	1 134	102 171
Pomoćni glagol	PG	2	327
Pridjev	A	13 854	1 114 605
Broj	M	69	863
Zamjenica	PRO	57	2 685
Prijedlog	S	80	80
Prilog	R	544	544
Veznik	C	88	88
Čestica	Q	54	54
Uzvik	I	17	17
Ostalo	NW	2	2
<b>UKUPNO</b>		<b>35 582</b>	<b>1 495 736</b>

Tablica 5 - broj vrsta riječi u glavnom leksikonu i broj generiranih oblika

U leksikonu **hrLexicon.dic** nalaze se samo osnovni oblici riječi. Uz svaku pojavnicu u leksikonu označena je i njezina leksička kategorija koja potpomaže pisanje gramatičkih pravila za prepoznavanje skupinskih kategorija (NP, PP, VP i AP).

**Imenice**, kojih ukupno ima 19681, međusobno su dodatno klasificirane pa se tako u leksikonu nalazi 141 kratika (N+kr), 353 vlastite imenice (N+vl), 19182 općih imenica (N+c) i 5 zbirnih imenica (N+r).

**Glagola** je ukupno 1134, a od toga ih je 376 povratnih glagola. Glagoli su nadalje klasificirani prema vidu na 561 svršeni glagol (Aspect=fin), 502 nesvršena (Aspect=inf) i 71 dvovidni glagol (Aspect=dual). Pomoćni glagoli *biti* i *htjeti* posebno su označeni i kao glavni glagoli (V) i kao pomoćni glagoli (PG).

**Pridjevi**, kojih je ukupno 13854, podjeljeni su na gradivne, posvojne i opisne pridjeve, ali i na određene i neodređene (u oblicima za koje je to bilo moguće).

U leksikonu je i 69 **brojeva** koji su dodatno klasificirani kao glavni, redni i zbirni brojevi, te brojevne imenice i brojevni pridjevi.

Od 57 **zamjenica** koje su u leksikonu, 6 je upitnih, 25 je neodređenih, 7 je ličnih, 8 je posvojnih, 9 je pokaznih, 1 povratno posvojna i 1 povratna.

Svaki prijedlog uz oznaku kategorije, ima još označene i padeže uz koje se pojavljuje. Tu sam informaciju koristila kod kreiranja lokalnih gramatika za prepoznavanje PP sintagmi.

### 3.3.2. Flektivne gramatike

Svi ostali oblici promjenljivih riječi generiraju se s pomoću opisa pojedinih klasa čije su paradigme navedene u vanjskim datotekama odnosno flektivnim gramatikama (\*.nof). Tako su paradigme za imenice opisane u datoteci hrCroNoun.nof, za pridjeve u hrCroAdj.nof, za glagole u hrCroVerb.nof, a za brojeve i zamjenice u hrCroNumPro.nof.

Veza na vanjske datoteke uspostavlja se preko naredbe #use hrCroNoun.nof u zaglavlju leksikona koji ih koristi. Kako bi pojava znała po kojem se opisu preoblikuje, uz nju se, osim oznake za vrstu riječi, obavezno navodi i oznaka za preobliku FLX=IME\_OZNAKE. Tako iz primjera koji slijede vidimo da je riječ '**mak**' označena kao opća (+c) imenica (N) muškog roda (+m) koja ima fleksiju prema paradigmi VLAK, glagol '**oduševiti**' ima fleksiju prema paradigmi CIJEPITI, pridjev '**zaljubljen**' prema paradigmi PRESPOR, broj '**četiri**' prema paradigmi TRI a zamjenica '**mi**' prema paradigmi MI.

✎ mak,N+c+m+FLX=VLAK

✎ oduševiti,V+FLX=CIJEPITI

✎ zaljubljen,A+op+nodr+FLX=PRESPOR

četiri,  $M+g1+FLX=TRI$

mi,  $PRO+l+FLX=MI$

U \*.nof datotekama nalaze se flektivni opisi koji se kompajliraju kao pretvarači konačnih stanja (engl. *finite state transducers*). Nakon generiranja svih oblika riječi iz leksikona, generira se minimalni deterministički pretvarač koji uz svaki oblik riječi navodi i njezin osnovni oblik te sve opise iz leksikona i iz flektivne gramatike.

Flektivne gramatike dopuštaju kreiranje ugniježđenih pravila koja se potom mogu umetnuti u bilo koju paradigmu. Pogledajmo kako je to napravljeno za imenicu MAK iz prethodnog primjera.

Imenica MAK u osnovnom leksikonu ima oznaku fleksije VLAK. U vanjskoj datoteci hrCroNoun.nof, ta je paradigma opisana na sljedeći način:

VLAK = :n060s + :n02p;

Oznake :n060s i :n02p same za sebe ništa ne znače. Tu se radi upravo o ugniježđenim pravilima čiji detaljniji opisi glase:

n060s = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>če/Voc+s + u/L+s + om/I+s;

n02p = ov(:n021p);

U opisu pravila n02p nalazimo još jedno ugniježđeno pravilo - :n021p koje izgleda ovako:

n021p = i/Nom+p + a/G+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p;

Iz navedenih pravila može se zaključiti da imenice, čija je oznaka za fleksiju VLAK, jedninu tvore tako da dodaju sljedeće nastavke na osnovni oblik: /, a, u, /, brisanje zadnjeg slova + če, u, om (pravilo n060s), dok množinu tvore nastavcima: i, a, ima, e, i, ima, ima (pravilo n021p) na način da prije samog nastavka dodaju niz 'ov' (pravilo n02p).

	imenice	glagoli	pridjevi	brojevi	zamjenice	ukupno
Ugniježđena pravila	23 s/ 20 m	25	6	0	4	<b>78</b>
Pravila	331	182	71	6	19	<b>609</b>
<b>Ukupno</b>	<b>374</b>	<b>207</b>	<b>77</b>	<b>6</b>	<b>23</b>	<b>687</b>

Tablica 6  
Pregled broja ugniježđenih pravila i punih pravila

Uvođenjem ugniježđenih pravila reducirala sam broj ponavljanja opisa paradigmi jer mnoge imenice imaju iste nastavke za tvorbu jednine ili množine, no razlikuju se po nizovima koji se dodaju prije samog nastavka (u primjeru za paradigmu VLAK dodaje se niz 'ov' prije nastavaka za tvorbu množine, dok, recimo, paradigma ALAT koristi te iste nastavke za množinu ali bez niza 'ov'). Popis i opis svih pravila za tvorbu imenica nalazi se u Prilogu 2.

Datoteka hrCroAdj.nof koristi ukupno 71 paradigmu za opis svih pridjeva koji se nalaze u leksikonu. I ovdje sam koristila ugniježđena pravila za redukciju opisa.

Tako većina pridjeva koristi Pozitiv paradigmu za opis pozitiv:

**Pozitiv** = a/G+m+s+np + u/D+m+s+np + a/Acc+m+s+np + i/Voc+m+s+np + u/L+m+s+np + im/I+m+s+np +

i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np + i/Voc+m+p+np + im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np

+ a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np + oj/L+f+s+np + om/I+f+s+np +

e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np + e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np

+ o/Nom+n+s+np + a/G+n+s+np + u/D+n+s+np + o/Acc+n+s+np + o/Voc+n+s+np + u/L+n+s+np + im/I+n+s+np +

a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np + a/Voc+n+p+np + im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np;

To se vidi i iz opisa paradigme PREZREO koja ima samo pozitiv:

**PREZREO** = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B1>l(:Pozitiv);

Za razliku od paradigme SVJEŽ koja uz pozitiv još ima i komparativ i superlativ:

**SVJEŽ** = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv  
 + (ij(:Komparativ)):KOM + ((ij(:Komparativ))<LW>naj):SUP;

Paradigma SVJEŽ tvori dakle oblike za pozitiv prema ugniježđenom pravilu **:Pozitiv**. Komparativ se tvori uz pomoć nastavaka koji su opisani u ugniježđenom pravilu **:Komparativ** na način da se prije nastavka, na osnovnu riječ doda 'ij' a potom se na svaki opis nastavka doda još i oznaka za komparativ koja je opisana ugniježđenim pravilom **:KOM**. Superlativ se tvori na isti način kao i komparativ s tom razlikom da se na početak svake riječi **<LW>** doda prefiks 'naj', a na kraju opisa pravila doda se oznaka za superlativ koja je opisana ugniježđenim pravilom **:SUP**.

**KOM** = <E>/+cp;

**SUP** = <E>/+sp;

**Komparativ** = (i/Nom+m+s + eg/G+m+s + ega/G+m+s + em/D+m+s + emu/D+m+s +  
 eg/Acc+m+s + i/Acc+m+s + ega/Acc+m+s + i/Voc+m+s + em/L+m+s + emu/L+m+s +  
 im/I+m+s +

i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + i/Voc+m+p +  
 im/L+m+p + ima/L+m+p + im/I+m+p + ima/I+m+p)

+ (a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + a/Voc+f+s + oj/L+f+s + om/I+f+s +

a/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + e/Voc+f+p + im/L+f+p +  
 ima/L+f+p + im/I+f+p + ima/I+f+p)

+ (e/Nom+n+s + eg/G+n+s + ega/G+n+s + em/D+n+s + emu/D+n+s + e/Acc+n+s +  
 e/Voc+n+s + em/L+n+s + emu/L+n+s + im/I+n+s +

a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + a/Voc+n+p +  
 im/L+n+p + ima/L+n+p + im/I+n+p + ima/I+n+p);

Popis i opis svih pravila za tvorbu pridjeva nalazi se u Prilogu 3.

Datoteka hrCroVerb.nof ima ukupno 182 paradigme koje su napravljene od raznih kombinacija 25 ugniježđenih pravila za tvorbu

infinitiva, prezenta, aorista, imperativa, imperfekta, glagolskog pridjeva radnog, glagolskog pridjeva trpnog, glagolskog priloga prošlog i glagolskog priloga sadašnjeg. Kao što se vidi iz sljedeća tri primjera pravila za paradigme VOLJETI, ZAPLAKATI i ŽIVJETI, i ovdje smo koristili ugniježđena pravila. Popis i opis svih pravila za tvorbu glagola nalazi se u Prilogu 4.

VOLJETI = :infinitiv + <B4>i(:PRe) + <B3>a(:IMP) + :PDRdoživjeti + <B2>n(:PDT) + <B4>i(:IMT) + <B4>eći/GPS;

ZAPLAKATI = :infinitiv + <B4>će (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B4>č(:IMT3) + <B2>vši/GPP;

ŽIVJETI = :infinitiv + <B4>i(:PRe) + <B4>lja(:IMP) + :PDRdoživjeti + <B4>i(:IMT) + <B4>eći/GPS;

Ostala složena glagolska vremena prikazana su uz pomoć glagolskih gramatika (vidi poglavlje 3.4.4.).

Datoteka hrCroNumPro.nof sadrži ukupno 6 paradigmi za opis brojeva i 23 paradigme za opis zamjenica (osobne, povratna, posvojne, pokazne, upitne i odnosne). Njihov potpuni pregled dan je u Prilogu 5 a predstavlja sažetiji zapis sustava za generiranje oblika hrvatskih riječi danoga u (Tadić, 1994.).

### 3.3.3. Opis svojstava

Leksikon hrLexicon.dic ima još jednu vezu na vanjsku datoteku **properties.def**. Naredbom #use properties.def unutar leksikona, stvara se veza između te dvije datoteke. U toj vanjskoj datoteci opisana su svojstva svih vrsta riječi koje se koriste u leksikonu.

Simbol '#' koristi se za pisanje komentara pa ga ovdje koristim kako bih objasnila značenje kratica koje sam koristila za opis svojstava.



**# NooJ V2 - Dictionary properties' definition**

# Language is: hr

**# Imenice - N**

N\_Case = Nom + G + D + Acc + Voc + L + I; #nominativ + genitiv + dativ + akuzativ + vokativ + lokativ + instrumental

N\_Gender = m + f + n; #muški + ženski + srednji rod

N\_Nb = s + p; # jednina + množina

N\_Type = c + vl + r + kr; #opće + vlastite + zbirne + kratice

**#Pomoćni glagoli - PG**

PG\_Tense = INF + INFk + PR + PRk + PRd + IMP + AO + AO2 + PDR + PDT + IMT + GPS + GPP + PER + FUT1 + FUT2;

# gl.vremena: infinitiv + prezent + imperfekt + aorist + gl.pridjev radni + gl.pridjev trpni + imperativ + gl.prilog sadašnji + gl.prilog prošli

PG\_Nb = s + p; # jednina + množina

PG\_Pers = 1 + 2 + 3; # 1, 2, 3 lice

PG\_Gender = m + f + n; #muški + ženski + srednji rod

**# Glagoli - V:**

V\_Tense = INF + INFk + PR + IMP + AO + PDR + PDT + IMT + GPS + GPP + PER + FUT1 + FUT2; # gl.vremena: infinitiv + infinitiv kratki + prezent + imperfekt + aorist + gl.pridjev radni + gl.pridjev trpni + imperativ + gl.prilog sadašnji + gl.prilog prošli

V\_Nb = s + p; # jednina + množina

V\_Pers = 1 + 2 + 3; # 1, 2, 3 lice

V\_Gender = m + f + n; #muški + ženski + srednji rod

V\_Type = pr + npr + pov + gbr; #prijelazni + neprijelazni + povratni + bez rekacije

V\_PAD = G + D + Acc + L + I; # za valenciju glagola

V\_Aspect = dual + inf + fin; # dvovidni + nesvršeni + svršeni

V\_DCobl = 0 + Nom + G + D + Acc + Voc + L + I + ind; # za valenciju glagola

V\_PCobl = 0 + Nom + G + D + Acc + Voc + L + I + ind; # za valenciju glagola

V\_DCtyp = 0 + Nom + G + D + Acc + Voc + L + I + ind; # za valenciju glagola

V\_PCtyp = 0 + Nom + G + D + Acc + Voc + L + I + ind; # za valenciju glagola

V\_SC = obl + typ; # za valenciju glagola

V\_AL = obl + typ; # za valenciju glagola

**#Brojevi - M:**

M\_Nb = s + p; # broj  
 M\_Gender = m + f + n; # rod  
 M\_Case = Nom + G + D + Acc + Voc + L + I; # padeži  
 M\_Type = gl + g + o + mlp + bi + bp; #glavni + redni + zbirni  
 # + brojevne imenice + brojevni pridjevi

**# Pridjevi - A:**

A\_Case = Nom + G + D + Acc + Voc + L + I; # padeži  
 A\_Gender = m + f + n; # muški + ženski + srednji rod  
 A\_Nb = s + p; # jednina + množina  
 A\_Com = np + cp + sp; # pozitiv + komparativ +  
 superlativ  
 A\_Type = op + gr + po; #opisni + gradivni + posvojni  
 A\_Form = odr + nodr; #određeni + neodređeni

**# Zamjenice - PRO:**

PRO\_Case = Nom + G + Gk + D + Acc + Acck + Voc + L + I; # padeži  
 PRO\_Gender = m + f + n; #muški + ženski + srednji rod  
 PRO\_Nb = s + p; # jednina + množina  
 PRO\_Type = l + psv + pvr + pvps + pkz + upt + odn + nodr;  
 # lične + posvojne + povratne + povratno-posvojna  
 # + pokazne + upitne + odnosne + neodređene

**#Prijedlozi - S:**

S\_Case = G + D + Acc + L + I; #padeži kojima otvaraju mjesta

**#Veznici - C**

C

**#Prilozi - R**

R

**#Čestice - Q**

Q

#Uzvici - I

I

INFLECTION =

+ INF + INFk + PR + PRk + PRd + IMP + AO + AO2 + PDR + PDT + IMT + GPS + GPP #  
glagolska vremena

+ s + p # broj

+ 1 + 2 + 3 # lice

+ m + f + n # rod

+ np + cp + sp; # komparacija pridjeva

Sva svojstva koja sam koristila u leksikonu i u flektivnim gramatikama, moraju biti navedena i na ovom mjestu. Svojstva se definiraju uz atribut (vrstu riječi) koji opisuju.

### 3.4. Sintaksna analiza

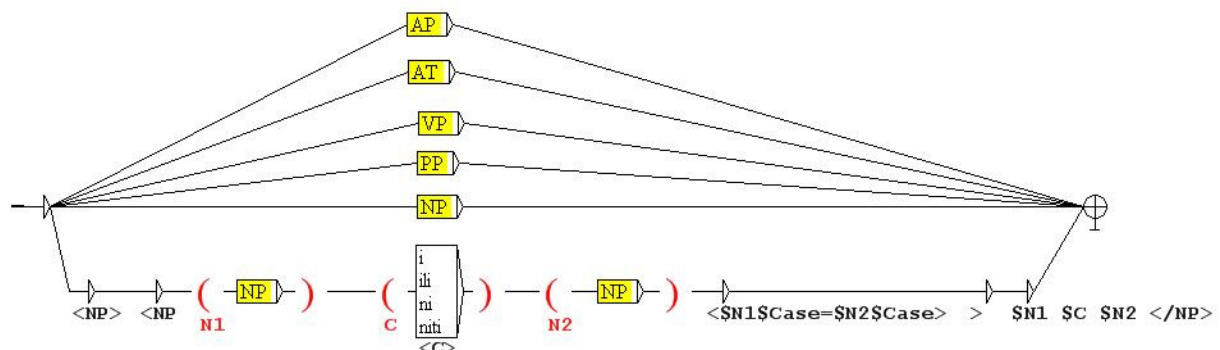
Sve gramatike koje sam pisala za izradu ovoga prvoga modela parsera za hrvatski jezik temelje se na modularnosti lokalnih gramatika. Svaka je gramatika pretvarač konačnih stanja koji se kaskadno povezuju dajući kao rezultat konačnu analizu rečenice kao osnovne jedinice za obradu. Broj gramatika koje koristim za sintaksnu analizu je 12 a primjenjuju se na tekst sljedećim redoslijedom:

- Brojevi.nog
- Nepromjenjive.nog
- GlagolJE.nog
- VP\_NP\_PP.nog
- obezlicenjeGlagola.nog
- VP\_DCObl\_PCobl.nog
- VP\_DCObl\_PCtyp.nog
- VP\_DC\_PC\_obl.nog
- VP\_DC\_PC\_typ.nog
- Biti\_htjeti.nog
- Subjekt\_predikat.nog
- Recenica.nog.

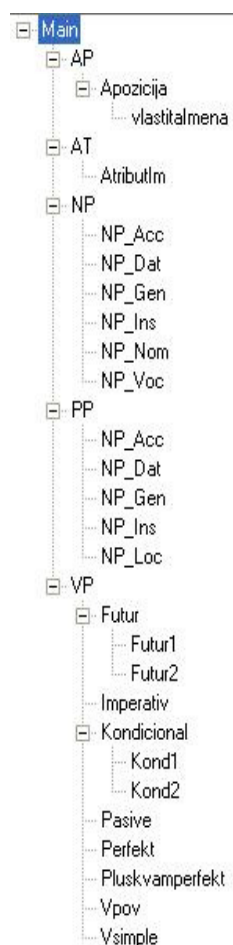
Gramatike će u tekstu biti opisane u sklopu opisa pripadajućih sintagmi, odnosno u sklopu poglavlja za opis uklanjanja višeznačnosti nad određenom sintagmom na način prikazan sljedećom tablicom.

	gramatika	vrsta analize	Poglavlje
1	Brojevi.nog	Razdjeljivanje	Brojevi
2	Nepromjenjive.nog	Razdjeljivanje	Nepromjenjive <VP> višeznačnosti
3	glagolJE.nog	razdjeljivanje	<VP> višeznačnosti
4	VP_NP_PP.nog	Plitko parsanje	VP-dio u Glagolska sintagma <VP>

			NP-dio	u
			Imenska	
			sintagma <NP>	
			PP-dio	u
			Prijedložna	
			sintagma <PP>	
			AP-dio	u
			Apozicijska	
			sintagma <AP>	
			AT-dio	u
			Atributska	
			sintagma <AT>	
5	Biti_htjeti.nog	Uklanjanje višeznačnosti	<NP>	
6	obezlicenjeGlagola.nog	Uklanjanje višeznačnosti	višeznačnosti	
7	VP_DCobl_PCobl.nog	Uklanjanje	Glagolska	
		višeznačnosti/prepoznavanje	sintagma <VP>	
		glagolskih dopuna	<NP>	
8	VP_Dcobl_Pctyp.nog	Uklanjanje	višeznačnosti	
		višeznačnosti/prepoznavanje	<NP>	
		glagolskih dopuna	višeznačnosti	
9	VP_DC_PC_obl.nog	Uklanjanje	<NP>	
		višeznačnosti/prepoznavanje	višeznačnosti	
		glagolskih dopuna		
10	VP_DC_PC_typ.nog	Uklanjanje	<NP>	
		višeznačnosti/prepoznavanje	višeznačnosti	
		glagolskih dopuna		
11	Subjekt_predikat.nog	Uklanjanje	<NP>	
		višeznačnosti/prepoznavanje	višeznačnosti	
		subjekt-predikat odnosa		
12	Recenica.nog	Prepoznavanje rečenica	Rečenica <S>	



Slika 27: Glavni graf VP\_NP\_PP.nog



Slika 28:  
Podgrafovi  
glavnog  
VP\_NP\_PP.nog  
grafa

Dobivena sintaksna analiza prikazana je linearno uz pomoć xml notacije. Gramatike su leksikalizirane što znači da koriste točno i samo one strukture koje su ovjerene prisutnošću u leksikonu. Na taj se način sve sintaksne i semantičke oznake pojedinih riječi u leksikonu automatski prenose na gramatički dopuštene sintagme.

Glavni graf VP\_NP\_PP.nog (vidi Sliku 27) prepoznaje:

- glagolske sintagme <VP>,
- imenske sintagme <NP>,
- atributne sintagme <AT>,
- apozicijske sintagme <AP> i
- prijedložne sintagme <PP>.

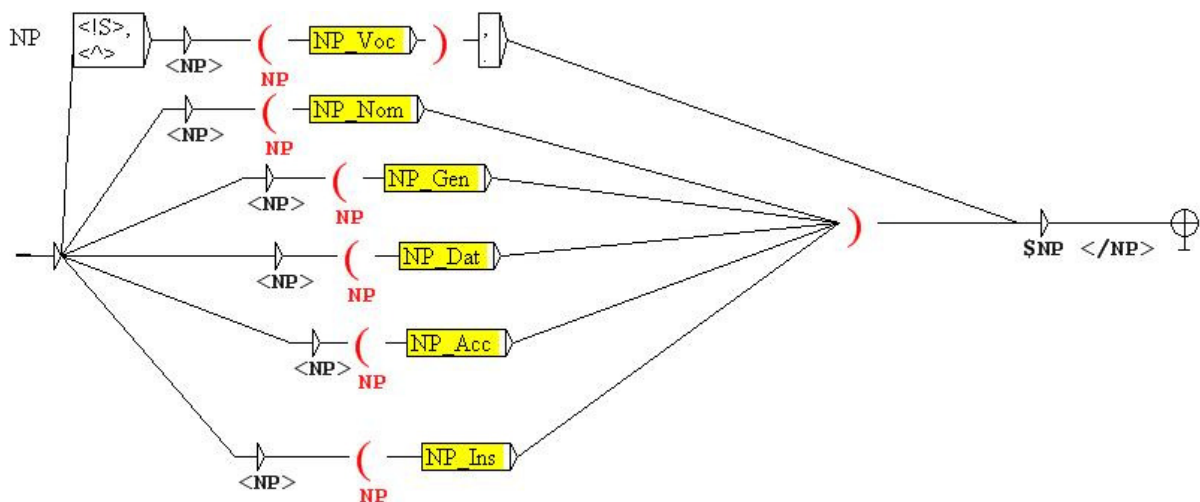
Sastoji se od ukupno 40 podgrafova (vidi Sliku 28) od kojih je:

- 5 glavnih podgrafova VP, PP, AT, AP i NP,
- 6 NP podgrafova za prepoznavanje nominativnih, genitivnih, dativnih, akuzativnih, vokativnih i instrumentalnih NP sintagmi,
- po 5 PP podgrafova za prepoznavanje genitivnih, dativnih, akuzativnih, lokativnih i instrumentalnih NP, AP i AT sintagmi unutar PP sintagmi,
- 6 AP podgrafova od kojih svaki ima po još jedan podgraf za opis vlastitih imena,
- 6 AT podgrafova, te, konačno,
- 12 VP podgrafova za prepoznavanje futura I. i futura II., imperativa, kondicionala I. i kondicionala II., pasiva, perfekta, pluskvamperfekta, jednostavnih i povratnih glagola i njihovih negacija.

Svi ostali grafovi bit će opisani u poglavlju Višeznačnosti jer služe upravo za uklanjanje mogućih višeznačnosti u analizi.

### 3.4.1. Imenska sintagma <NP>

Imenska sintagma <NP> sadrži imenicu kao glavu sintagme s njezinim dodatnim opisom. U tom opisu mogu biti, u bilo kojoj kombinaciji, brojevi, zamjenice i pridjevi koji se s imenicom slažu u rodu, broju i padežu. Glavna gramatika za prepoznavanje imenskih sintagmi je gramatika NP i ona sadrži šest podgramatika (vidi Sliku 29).

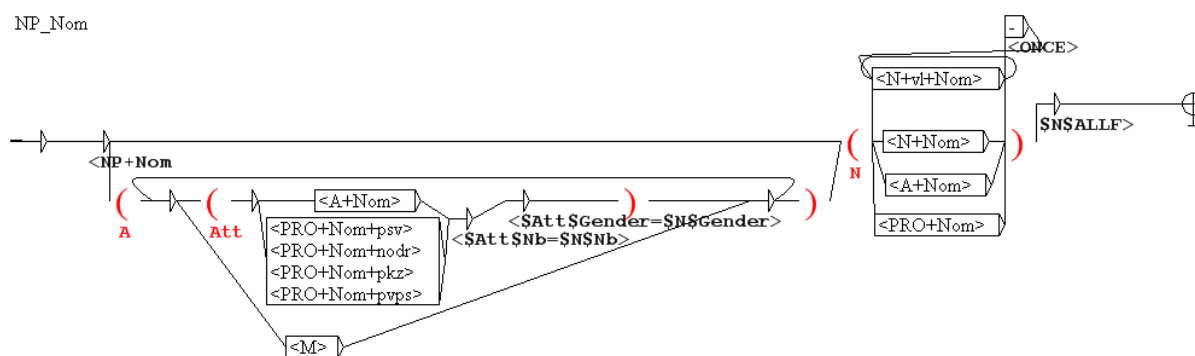


Slika 29: Osnovni <NP> graf

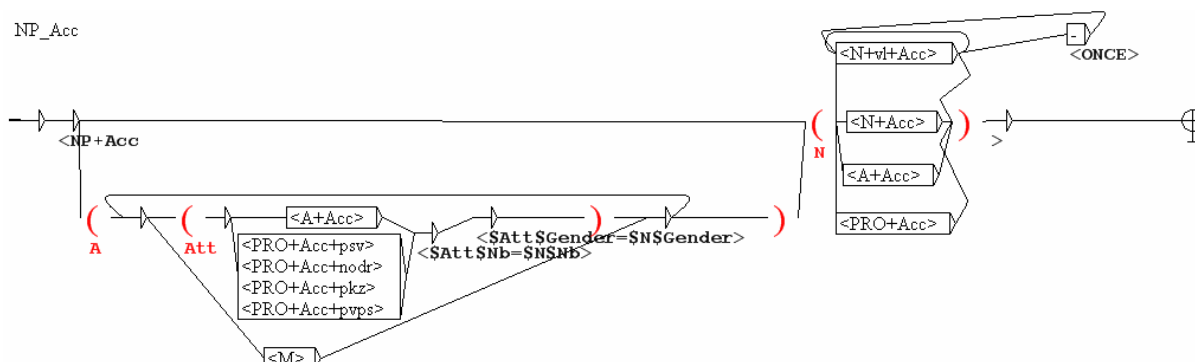
U gramatici ne postoji podgraf za opis <NP> sintagme u lokativu. Tome je razloga taj što padež lokativ dolazi isključivo uz pojavu lokativnih prijedloga u rečenici pa se on nalazi samo u gramatici za opis <PP> sintagmi. Grana za opis <NP> sintagmi u vokativu, dodatno je opisana jednim čvorom prije i poslije samog <NP+Voc> čvora. Na taj se način ograničava obilježavanje vokativnih NP struktura na samo one ispred kojih nema ništa, tj. nalaze se na početku rečenice ili ispred njih nije prijedlog sa zarezom dok iza njih mora doći ili kraj rečenice ili zarez.

U svakoj podgramatici opisano je da se <NP> sintagma sastoji ili od jedne opće imenice, vlastite imenice, pridjeva ili zamjenice ili od više vlastitih imenica ili od neograničenog broja pridjeva, zamjenica (pokaznih, neodređenih, posvojnih ili povratno-posvojnih) i brojeva ispred imenice, pridjeva ili zamjenice, a koji se međusobno slažu u rodu, broju i padežu. Gramatike za nominativ, genitiv, dativ, akuzativ, vokativ i instrumental

međusobno se razlikuju samo u oznaci padeža unutar svakog čvora kao što se vidi iz primjera za nominativ (vidi Sliku 30) i primjera za akuzativ (vidi Sliku 31).



Slika 30: Primjer za <NP> u nominativu



Slika 31: Primjer za <NP> u akuzativu

U svom osnovnom obliku, gramatika NP je dala 1594 konkordancije. Neke od njih su:

- ✍ 90 radnih mjesta
- ✍ adekvatnim rješenjima
- ✍ bivši predsjednički kandidat
- ✍ današnji ukupan položaj



- ✎ *dramatične beogradske promjene*
- ✎ *hrvatske vanjske politike*
- ✎ *obećanih 55 milijuna dolara*
- ✎ *svoje profesionalne sposobnosti*

Treba napomenuti da je gramatika prepoznavala i riječi poput:

- ✎ *ali (N=Ali i N=ala), barem (N=Bare), biti (N=bit), bila (N=bilo),*
- ✎ *donosi (N=donos), kada (N=kada), koliko (N=kolika),*
- ✎ *moći (N=moć), oda (N=oda), odana počast (A=odan),*
- ✎ *pače (N=pače), pod (N=pod), prije (N=prija),*
- ✎ *prođe (N=prođa), znače (N=znak).*

Neke od njih sam pokušala opisati i dodatnim gramatikama (vidi poglavlje o višeznačnostima). Osim riječi koje ne pripadaju imenskoj sintagmi, neke su se imenske sintagme ponavljale više puta zbog međusobnog istog lika u više padeža, npr.

- ✎ *puko tapšanje <NP+Nom> <NP+Acc>*
- ✎ *Demokratski centar <+Nom+Acc>*
- ✎ *pravi izazov <+Nom+Acc>*
- ✎ *takve vrste <+Nom+Acc+G>*
- ✎ *Srpske narodne stranke <+Nom+Acc+G>*
- ✎ *Hrvatske Vlade <+Nom+Acc+G>*
- ✎ *Lokalne samouprave <+Nom+Acc+G>*
- ✎ *Hrvatske vanjske politike <+Nom+Acc+G>*

✍ Glavnog urednika <+Acc+G>

✍ Generalnog direktora <+Acc+G>

✍ Brojnim ministrima <+D+I>

Uglavnom se radi o preklapanju oblika za nominativ, akuzativ i genitiv, te dativ i instrumental koji nastaju zbog izraženoga sinkretizma padeža u hrvatskome jeziku.

U sklopu imenske sintagme opisala sam i koordinaciju, odnosno dvije imenske sintagme u istom padežu međusobno povezane veznicima 'i', 'ili', 'ni' ili 'niti'. Prikaz njihove gramatike vidljiv je na slici 'glavni graf VP\_NP\_PP.nof'.

Dvije sintagme koje slijede zapravo su potkategorije imenske sintagme pa su u analizi označene kao <NP TYPE=AP> i <NP TYPE=AT>. No, s obzirom na njihovu specifičnost, smatram da zaslužuju poseban oblik obradbe.


### 3.4.2. Apozicijska sintagma <AP>

Graf za prepoznavanje apozicija (vidi Sliku 32) razlikuje se od prethodnih grafova za opis imenskih sintagmi. Naime, u ovoj se podgramatici traže dvije imenice koje stoje jedna uz drugu, a slažu se u padežu i broju. Obje imenice mogu biti opće imenice, ili prva može biti opća imenica i skraćenica, a druga je vlastito ime unutar kojeg se jedanput može pojaviti i vezna crtica (vidi Sliku 33).



Iako u apozicijskoj sintagmi, svaka od imenica može imati još i svoje atribute, tj. neograničen broj pridjeva, zamjenica i brojeva koji se slažu u rodu, broju i padežu s imenicom uz koju stoje (u ovom slučaju, uz glavnu imenicu i uz apozicijsku imenicu) za sada sam opisala samo pojavljivanje jednog atributa prije imenice (vidi Sliku 34).

Primjeri koji su prepoznati ovom gramatikom su:



↗	<i>psa skitnicu</i>
↘	<i>ptice selice</i>
↗	<i>misao vodilja</i>
↘	<i>grofu Marku</i>
↗	<i>starac Ante</i>
↘	<i>zamjenica Vesna Škare-Ožbolt</i>

### 3.4.3. Atributska sintagma <AT>

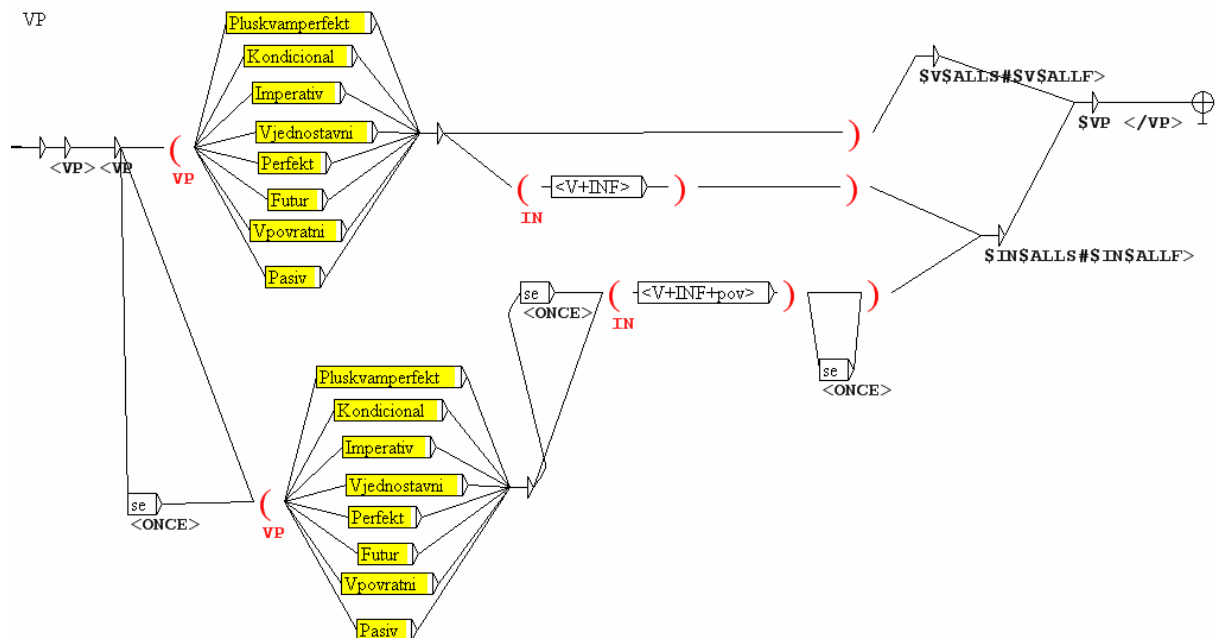
Slično kao i kod apozicijske sintagme, gramatika za prepoznavanje imenskog atributa traži dvije imenice koje se ovaj put ne moraju slagati u padežu. Uvjet koji treba biti zadovoljen je da je druga imenica u nizu u genitivu. Isto vrijedi i za pridjev ili posvojnu zamjenicu koje joj mogu biti dopuna i koje se s imenicom još moraju slagati u rodu i broju. Također, imenica koja je atribut u ovoj sintagmi, može imati svoj pridjev kao dopunu s kojim se mora slagati u rodu, broju i padežu (vidi Sliku 36). Gramatika za prepoznavanje imenskog atributa (vidi Sliku 35) opisana je na sljedeći način:



### 3.4.4. Glagolska sintagma <VP>

Glagolsku sintagmu <VP> može činiti jedan jednostavni glagolski oblik, složeni glagolski oblik ili povratni jednostavni ili složeni glagolski oblik, kao i negacije bilo kojeg od tih oblika. Kod složenih glagolskih vremena opisane su sve kombinacije odnosa glavnog glagola, povratnog glagola, povratne zamjenice 'se' i negacije koje su ovjerene za hrvatski jezik.

Početna VP gramatika ima glavni graf (vidi Sliku 37) koji se sastoji od 8 podgrafova. Podgrafovi za futur i kondicional se nadalje sastoje od po još dva potpodgraфа što u konačnici s glavnim grafom čini ukupno 12 grafova za označavanje glagolskih skupina.

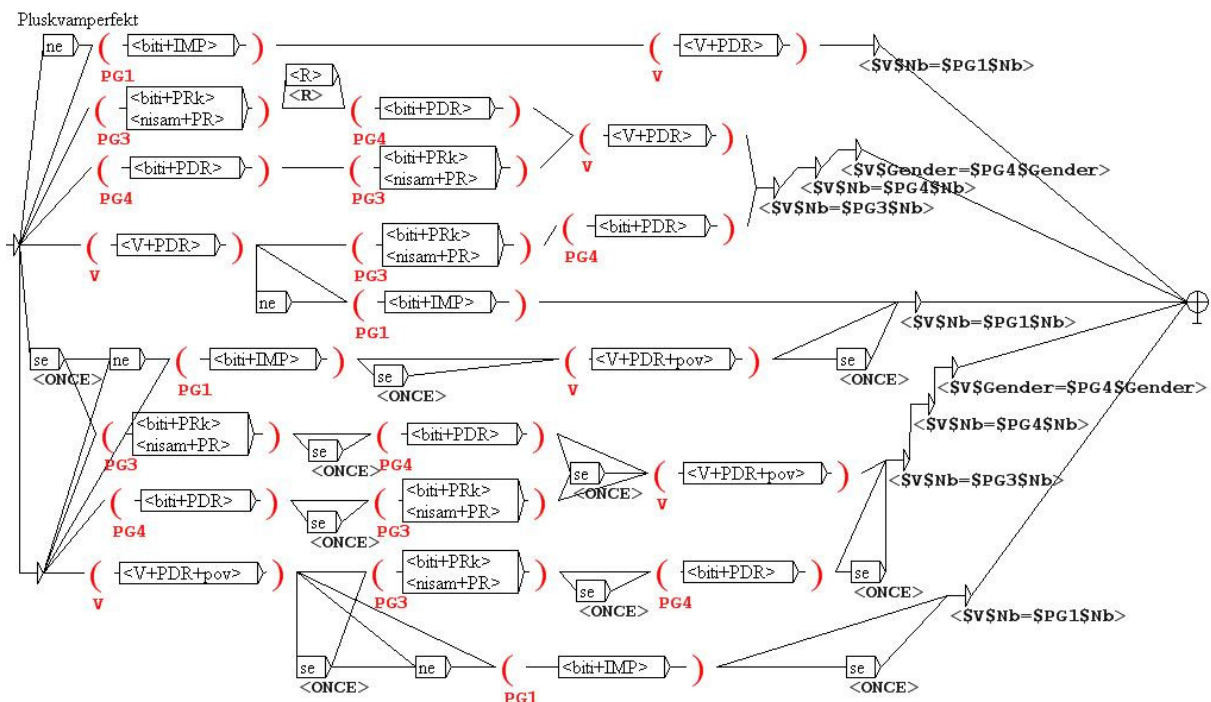


Slika 37: Glavni VP graf

Glavni VP graf ima granu za opis glagolskih oblika povratnih i nepovratnih glagola, zatim opis glagolskih oblika nepovratnih glagola koji u svojoj sintagmi mogu imati infinitiv na kraju tj. modalnih glagola i konačno opis glagolskih oblika s povratnim infinitivom na kraju sintagme. Dodatna grana za opis posljednjeg slučaja, dakle, kad se radi o konstrukciji u kojoj je infinitiv povratni glagol, bila je potrebna zbog različitog mjesta unutar <VP> sintagme na kojem se može pojaviti zamjenica 'se', npr.:



Nakon prepoznavanja glagolskog vremena koje nema infinitiv na svom kraju, cijeli VP izraz preuzima sve sintaksne i flektivne oznake glavnog glagola sintagme. No, ako se na kraju glagolske sintagme nalazi i glagol u infinitivu, tada cijeli VP izraz preuzima njegove sintaksne i flektivne oznake. Preuzimanje ovih oznaka potrebno je za opis gramatike kojom prepoznajem subjekt i objekt kojima glavni glagol otvara mjesta u rečenici. Slijedi opis grafova za prepoznavanje navedenih glagolskih vremena.



Slika 38: Graf za prepoznavanje pluskvamperfekta povratnih i nepovratnih glagola

Gramatika za prepoznavanje pluskvamperfekta (vidi Sliku 38) prepoznaje sve kombinacije tog složenog glagolskog vremena uključujući i

<sup>95</sup> Primjeri koji završavaju sa zvjezdicom označavaju stilski obilježen red riječi.

njegove moguće negacije kao i položaj povratne zamjenice 'se' u slučaju povratnih glagola, npr.:

✎ ...bijah pjevao...

✎ ...bio sam pjevao...

✎ ...pjevao sam bio...

✎ ...pjevao bijah...

✎ ...ne bijah pjevao...

✎ ...pjevao ne bijah...

✎ ...nisam bio pjevao...

✎ ...pjevao nisam bio...

Gramatika za prepoznavanje kondicionala sastoji se od dva podgrafa za prepoznavanje kondicionala I. (vidi Sliku 39) i kondicionala II (vidi Sliku 40).

Ovom su gramatikom opisani primjeri poput:

✎ ...bih pjevao...

✎ ...pjevao bih...

✎ ...ne bih pjevao...

✎ ...pjevao ne bih...

✎ ...bih se smijala...

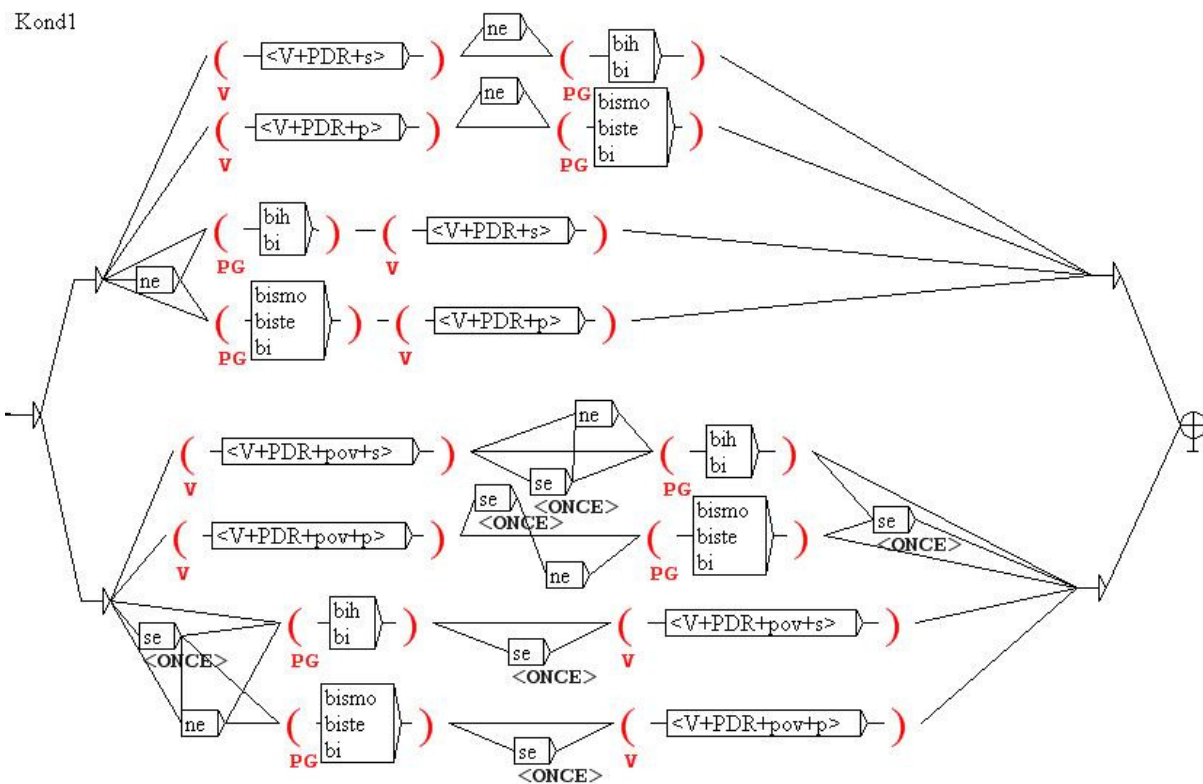
✎ ...smijala bih se...

✎ ...ne bih se smijala...

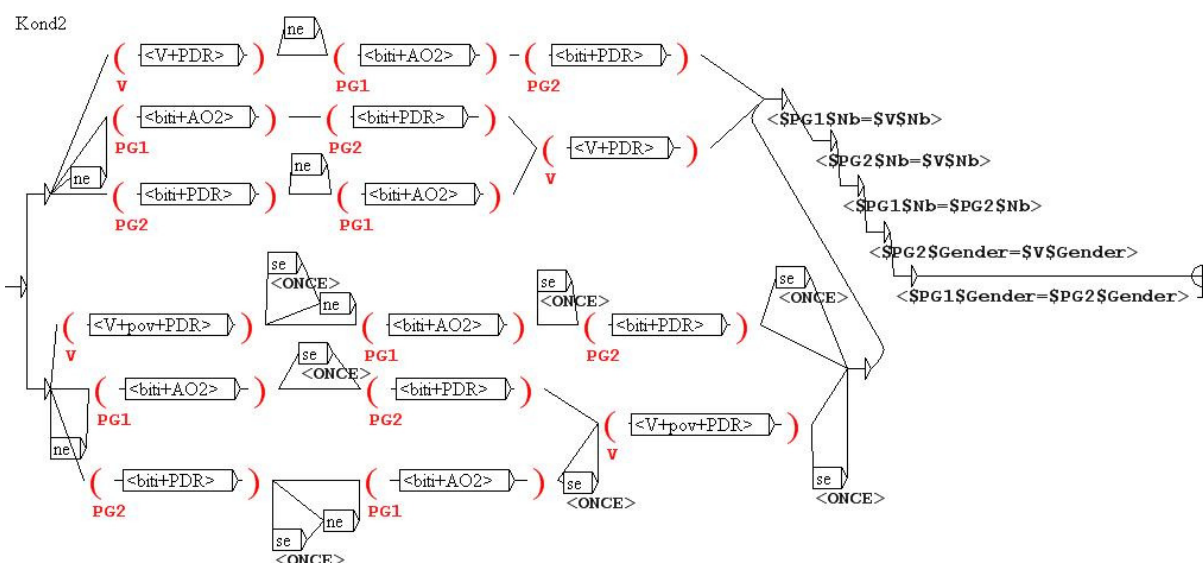
✎ ...smijala se ne bih...



✍ ...se ne bih smijala



Slika 39: Graf za prepoznavanje kondicionala I. povratnih i nepovratnih glagola



Slika 40: Graf za prepoznavanje kondicionala II. povratnih i nepovratnih glagola

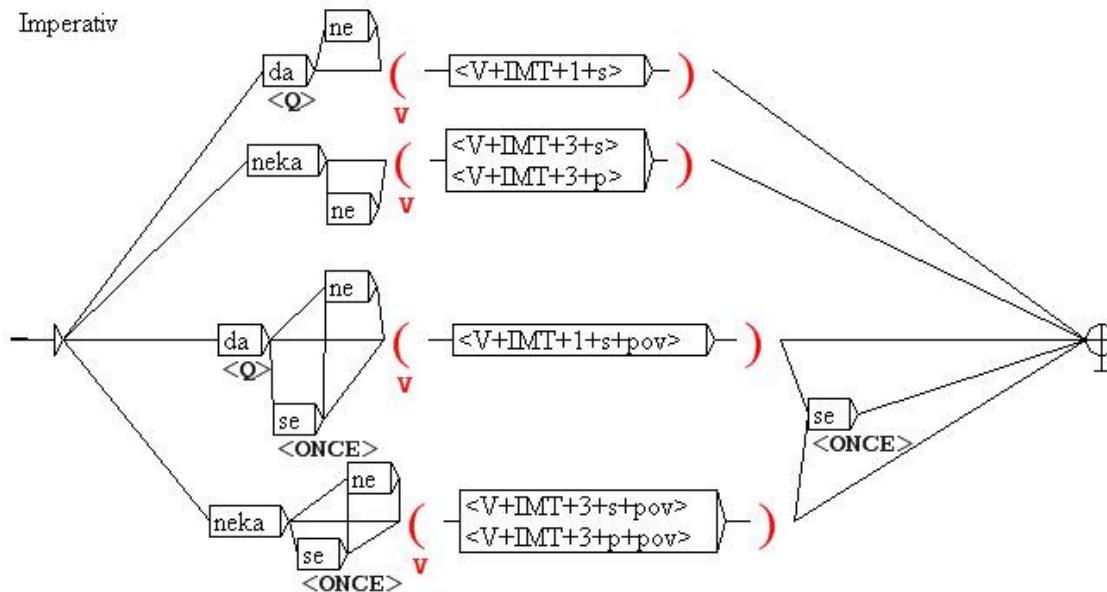
Graf za prepoznavanje oblika u kondicionalu II. prepoznaje sljedeće kombinacije:

	<i>bio bih pjevao</i>
	<i>bih bio pjevao</i>
	<i>pjevao bih bio</i>
	<i>pjevao ne bih bio</i>
	<i>ne bih bio pjevao</i>
	<i>bio ne bih pjevao</i>
	<i>bio bih se smijao</i>
	<i>smijao bih se bio</i>
	<i>smijao bih bio se*</i>
	<i>smijao se ne bih bio</i>
	<i>smijao ne bih se bio</i>
	<i>ne bih se bio smijao</i>
	<i>bio se ne bih smijao*</i>

Oblici glagola za imperativ već su opisani u sklopu flektivnih gramatika svakog glagola. Zato su ovdje dodatno opisani još samo prvo lice jednine i treće lice jednine i množine jer su, za razliku od drugoga lica, složeni od glavnog glagola i riječi 'da' za prvo lice jednine kao što je opisano u (Silić, Pranjković, 2005:74) i riječi 'neka' za treće lice jednine i množine (vidi Sliku 41).

	<i>Ajde da završim s tim.</i>
	<i>On neka završi s tim.</i>

Gramatika još opisuje i različite mogućnosti vezanja negacije za opisana lica dok je drugo lice dodatno opisano u zasebnoj gramatici s ostalim jednostavnim oblicima glagolskih vremena.



Slika 41: Graf za prepoznavanje imperativa povratnih i nepovratnih glagola

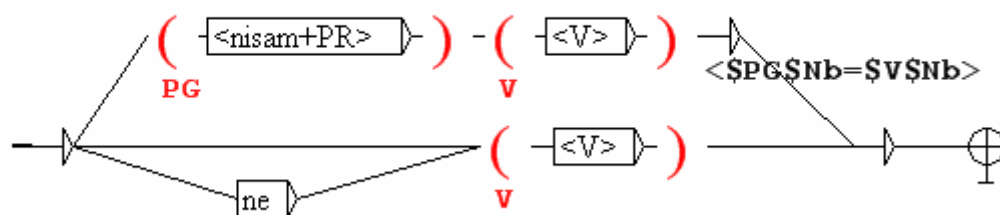
Ovaj graf prepoznaje sljedeće oblike:

- ✂ neka pjeva
- ✂ neka ne pjeva
- ✂ neka se smije
- ✂ neka se ne smije
- ✂ neka ne smije se<sup>96</sup>

Slijedi graf koji prepoznaje sve jednostavne glagolske oblike (prezent, aorist, imperfekt, 2. lice jednine i 1. i 2. lice množine imperativa, glagolski prilog prošli i sadašnji) i njihove negacije (vidi Sliku 42).

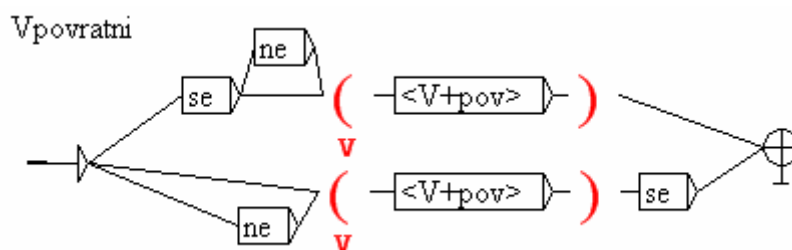
<sup>96</sup> Stilski obilježen red riječi.

Vjednostavni



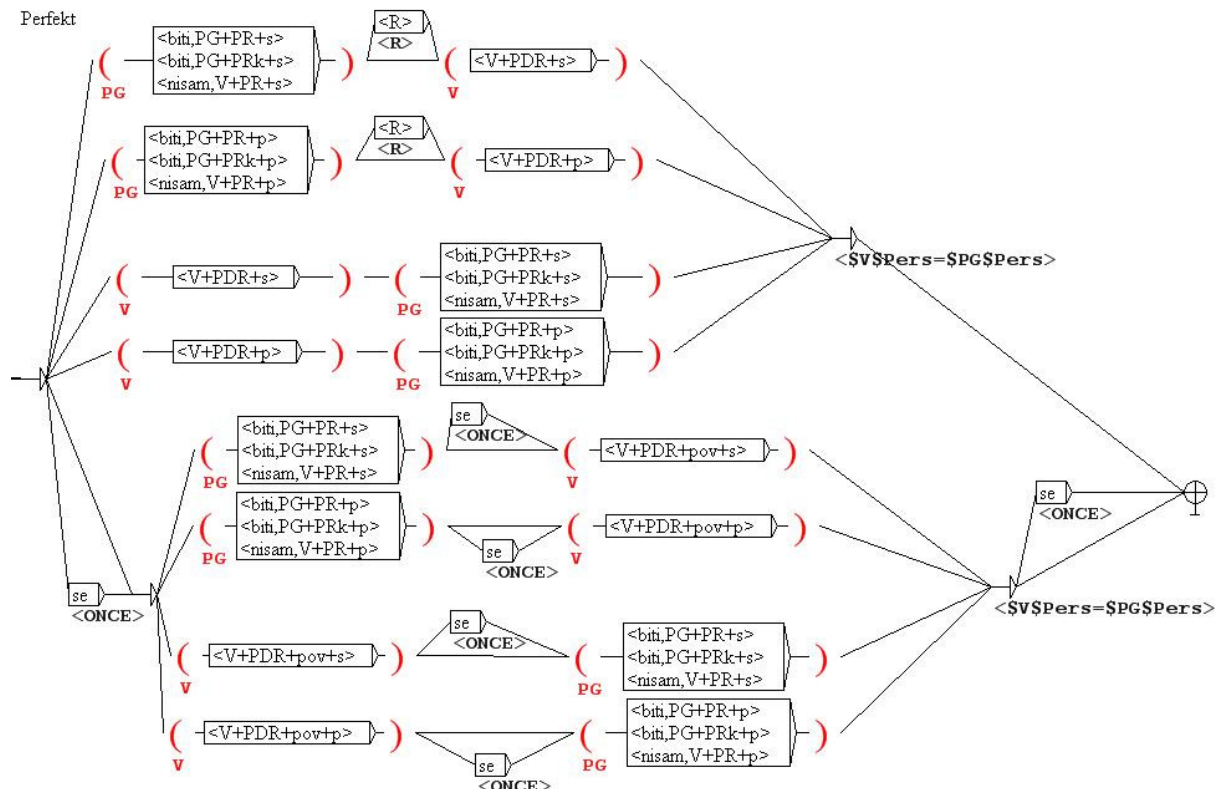
Slika 42: Graf za prepoznavanje jednostavnih glagolskih oblika

Prepoznavanje jednostavnih povratnih glagolskih oblika i njihove negacije opisano je zasebnom gramatikom (vidi Sliku 43).



Slika 43: Graf za prepoznavanje jednostavnih glagolskih oblika povratnih glagola

Gramatika kojom se prepoznaje perfekt povratnih i nepovratnih glagola i njihove negacije, opisana je sljedećim grafom (vidi Sliku 44).



Slika 44: Graf za prepoznavanje perfekta

Ovom gramatikom prepoznaju se oblici poput:

je pjevao

pjevao je

nije pjevao

pjevao nije

se je smijao\*

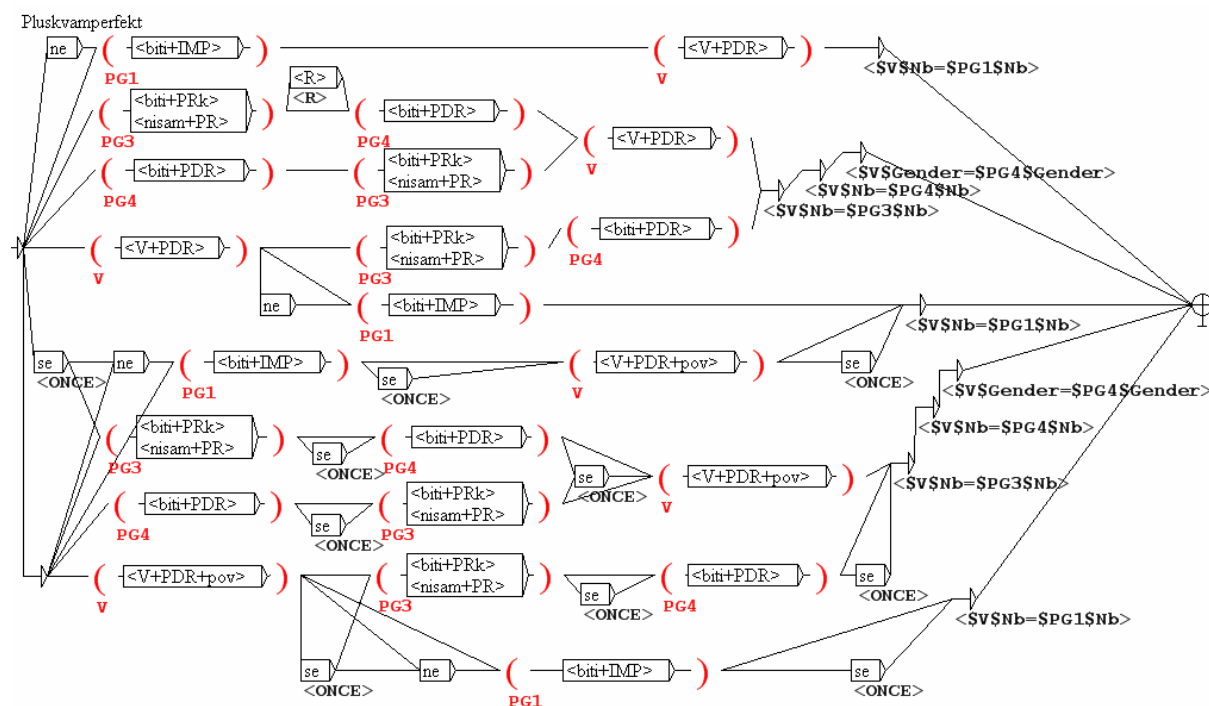
smijao se je

se nije smijao

se smijao nije\*

nije se smijao

















Posljednja gramatika za opis glagolskih oblika prepoznaje povratne i nepovratne glagole kao i njihove negacije u pluskvamperfektu (vidi Sliku 45).



Slika 45: Graf za prepoznavanje pluskvamperfekta

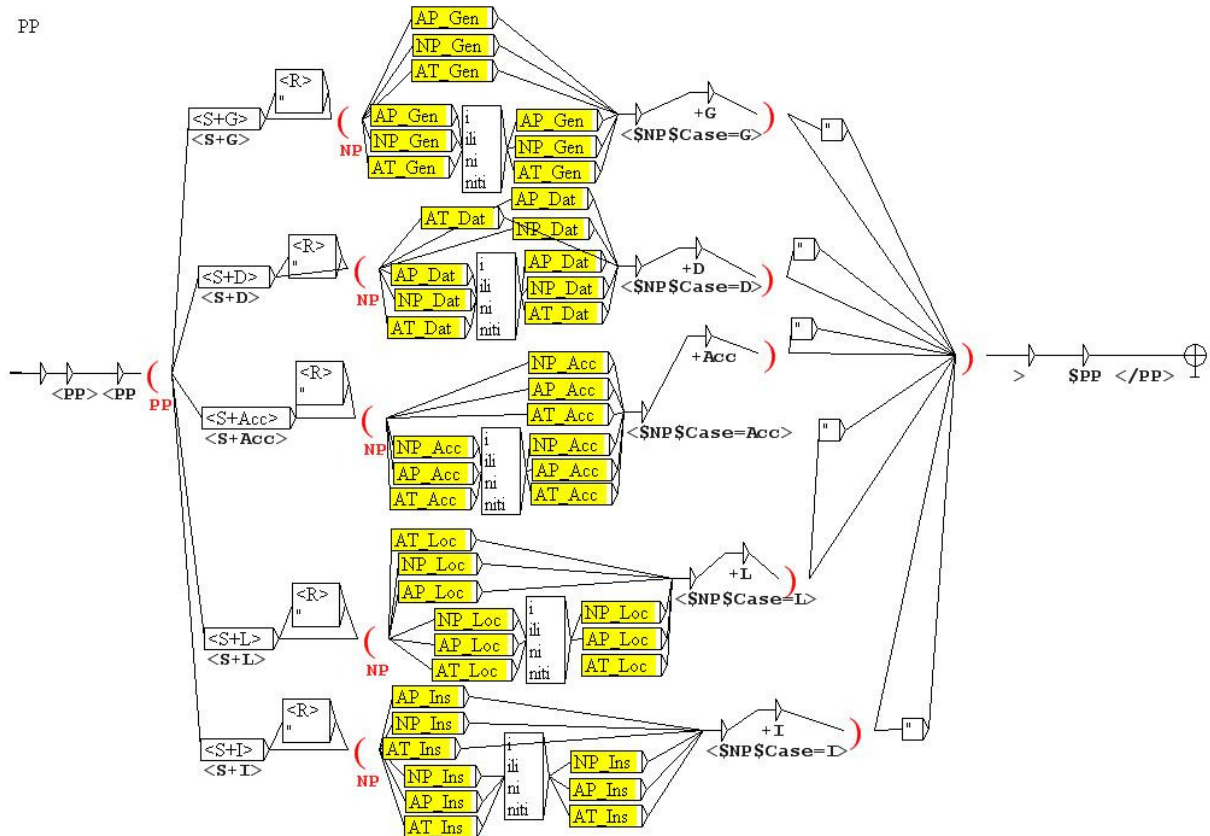
Ovom gramatikom prepoznaju se oblici poput:

✂	<i>pjevao bio sam</i>
✂	<i>bio sam pjevao</i>
✂	<i>sam bio pjevao</i>
✂	<i>pjevao bijah</i>
✂	<i>bijah pjevao</i>
✂	<i>ne bijah pjevao</i>
✂	<i>nisam bio pjevao</i>
✂	<i>bio nisam pjevao</i>

	 <i>pjevao nisam bio</i>
	 <i>pjevao ne bijah</i>
	 <i>se ne bijah smijao*</i>
	 <i>ne bijah se smijao</i>
	 <i>ne bijah smijao se</i>
	 <i>nisam se bio smijao</i>
	 <i>se nisam bio smijao</i>
	 <i>nisam bio se smijao*</i>
	 <i>bio se nisam smijao*</i>
	 <i>bio nisam se smijao*</i>
	 <i>bio nisam smijao se*</i>
	 <i>smijao se ne bijah</i>
	 <i>smijao ne bijah se*</i>
	 <i>smijao se nisam bio*</i>
	 <i>smijao nisam se bio</i>
	 <i>smijao nisam bio se*</i>

### 3.4.5. Prijedložna sintagma <PP>

Prijedložna sintagma ima prijedlog kao glavu skupa i imensku sintagmu kao svoju dopunu. Prijedložna sintagma je u stvari <NP>, <AP> ili <AT> ispred kojih se nalazi prijedlog uz uvjet da se padež NP-a, AP-a i AT-a slaže s padežom prijedloga, što se vidi i na grafu (vidi Sliku 46).



Slika 46: Glavni PP graf

Kao što se je prikazano grafom, nakon prijedloga može doći ili jedan NP, jedan AP ili jedan AT koji se s prijedlogom slažu u padežu. No, isto tako, iza prijedloga se mogu naći dva NP-a, dva AP-a, dva AT-a, ili bilo koja njihova kombinacija. Jedina dva uvjeta koja moraju biti zadovoljena su da su obje sintagme u istom padežu i da su međusobno vezana veznicima 'i', 'ili', 'ni' ili 'niti'. Moguće je da se između prijedloga i sintagmi nađe i jedan prilog.

S obzirom da se ispred nominativa i vokativa ne pojavljuje prijedlog, te kombinacije nismo uvrstili u ovu gramatiku.

Neki od primjera koji su prepoznati ovom gramatikom su:

u tim okvirima

o nemoćnoj paralizi

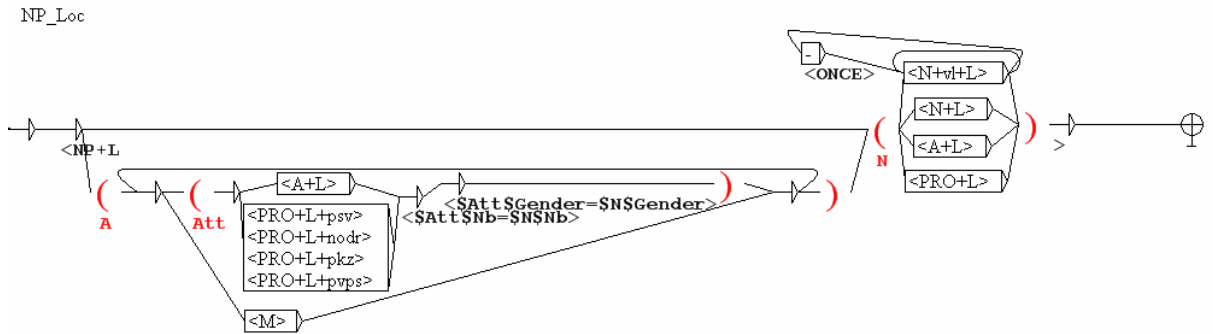
o stabilnosti

u prvom redu



	✂ na ručku
	✂ u Jonsko more
	✂ za novu istočnu politiku
	✂ nakon ovakvog Bruxellesa
	✂ s mnogo dodatnog strpljenja
	✂ od 185 milijuna dolara
✂	oko 20 milijuna njemačkih maraka
✂	iz hrvatskih državnih struktura
	✂ ispod državnog prosjeka
	✂ između Budimpešte i Jadrana
✂	između Hrvatske i međunarodne zajednice
	✂ na domaće i strane ulagače
✂	na onom utemeljiteljskom sarajevskom skupu
	✂ nakon jednoipolsatnog sastanka
	✂ o izuzetno povoljnim sredstvima
✂	o poticajnim mjerama i poreznim povlasticama
	✂ od najvećih svjetskih koncerna
	✂ za neke ekološke projekte

Detaljniji opis <NP> podgrafova ove gramatike dan je u poglavlju o Imenskim sintagmama. Jedini podgraf koji se tamo ne nalazi je onaj za prepoznavanje lokativa koji je identičan podgrafovima za prepoznavanje ostalih padeža (vidi Sliku 47).

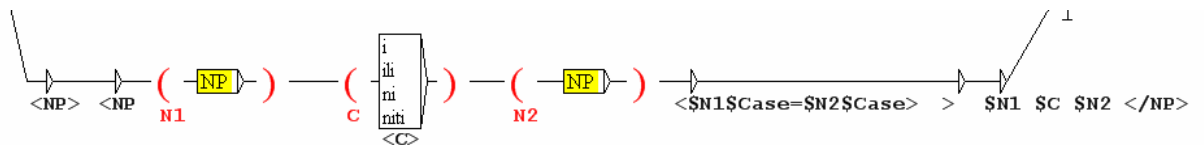


Slika 47: Graf za prepoznavanje &lt;NP&gt; dijela &lt;PP&gt; sintagme u lokativu

Isti je slučaj s gramatikama u kojima se iza prijedloga nalazi atributna sintagma ili apozicijska sintagma, pa su i one (<AP+Loc> i <AT+Loc>) morala biti dodatno opisane unutar glavne gramatike za prepoznavanje PP sintagmi.

### 3.4.6. Koordinacija

Na početnom grafu (Slika 27) postoji još jedna grana kojom je opisana koordinaciju dvije imenske sintagme (vidi Sliku 48). Na taj su način povezane dvije prepoznate imenske sintagme u jednu imensku sintagmu, ako se one slažu u padežu i ako su povezane veznicima 'i', 'ili', 'ni', 'niti'.

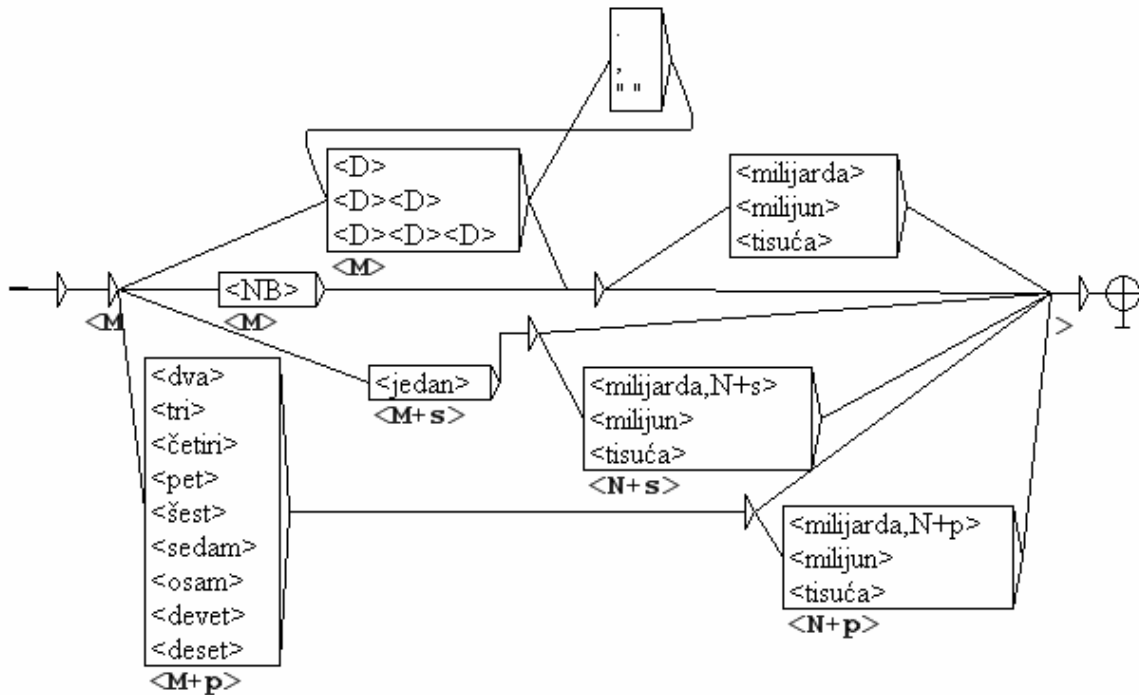


Slika 48: Dio grafa VP\_NP\_PP.nog

Koordinacija unutar prijedložne sintagme, opisana je u sklopu glavnog PP grafa (Slika 33).

### 3.4.7. Brojevi <M>

S obzirom da je bilo nemoguće (vremenski dugotrajno, a istovremeno i nepotrebno) u leksikonu navesti sve moguće brojeve u hrvatskom jeziku pisane slovima ili kombinacijom brojki i slova, napisala sam gramatiku kojom se prepoznaje veći dio brojnih kombinacija (vidi Sliku 49).



Slika 49: Brojevi

Ovom gramatikom prepoznaju se primjeri poput:

✍	10 milijardi
✍	7
✍	1.1 <sup>97</sup>
✍	111,11
✍	1.111,11

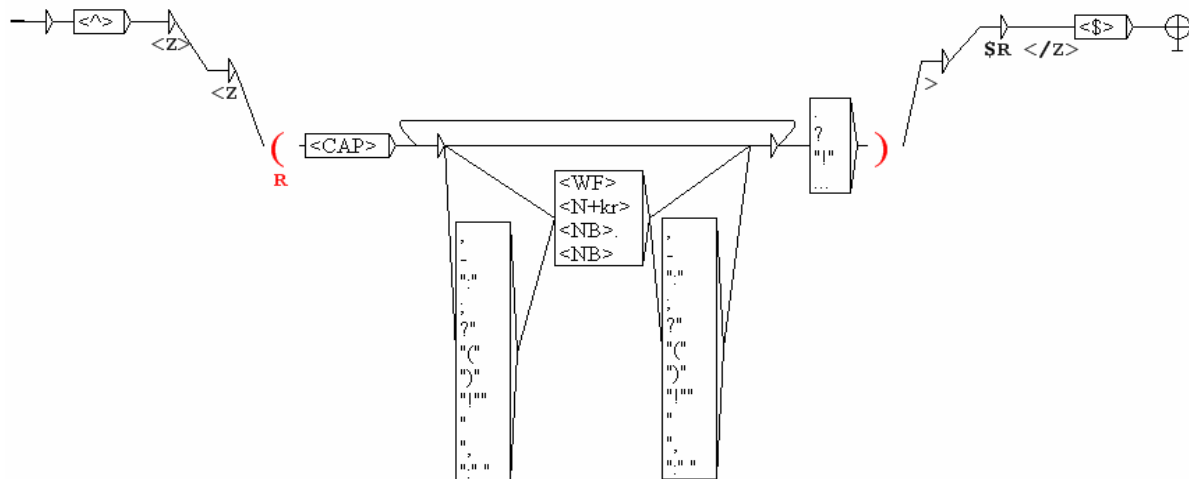
<sup>97</sup> Iako se u hrvatskome jeziku koristi decimalni zarez a ne decimalna točka, u tekstovima se često nalaze i jedan i drugi tip zapisa kao što je i vidljivo iz ovog primjera.

	✍ 1 milijarda
	✍ 2 milijarde
	✍ 21 tisuću
	✍ 5,5 milijuna
	✍ jedna tisuća
	✍ dvije tisuće
	✍ devet milijuna

Redne brojeve pisane slovima nisam ovdje opisivala jer je njihov opis dao Bekavac u svom doktorskom radu (Bekavac, 2005.)

### 3.4.8. Rečenica <S>

Gramatika za opis rečenice je veoma jednostavna (vidi Sliku 50). Rečenica započinje riječju koja počinje velikim slovom, a iza koje može doći ili kraj rečenice ili neograničen broj riječi <WF>, brojeva <NB>, brojeva s točkom <NB>., kratica s točkom <N+kr>, a između svakih od njih može se pojaviti jedan od interpunkcijskih znakova , - : ; ? » ! » : » ) ( » « .



Slika 50: Gramatika za opis rečenice

Kao što sam već objasnila u poglavlju 3.1.1. u ovaj sam rad uključila parser za analizu samo jednostavnih rečenica. Složene rečenice ostavljam za neko buduće i mnogo opsežnije istraživanje.

### 3.4.9. Znakovi interpunkcije

Istraživanja koja su proveli (Engel, Charniak, Johnson, 2002.; Roark<sup>98</sup>, 2001, Gregory, Johnson, Charniak, 2007., Jones, 1994.) pokazala su da prisutnost osnovnih interpunkcijskih znakova (točka, zarez, upitnik, uskličnik) poboljšava točnost<sup>99</sup> parsanja. Isto čini i odsutnost upadica (hm, ah, ma da, ovaj, znaš...) i unutrašnjih rečeničnih prozodija (kodirane pauze).

Za znakove interpunkcije nisam pisala posebne gramatike, ali sam za opis nekih drugih sintagmi koristila i znakove interpunkcije. Tako je na primjer, za opis gramatike kojom se u tekstu prepoznaju vokativi korišten zarez, točke su korištene za prepoznavanje datuma i kratica, crtica za prepoznavanje padežnih nastavaka za neke kratice i u slučajevima vezanja dvaju prezimena. Dodavanjem ovih znakova u sam opis gramatike, povećana je učinkovitost parsera.

### 3.4.10. Višeznačnosti

U ovom ću poglavlju detaljnije opisati sve preostale grafove koji većinom služe za uklanjanje višeznačnosti.

---

<sup>98</sup> Roark, B.: *Robust Probabilistic Predictive Syntactic Processing: Motivations, Models, and Applications*. Doktorska disertacija, Department of Cognitive Science, Brown University, Providence, 2001.

<sup>99</sup> Gregory, Johnson i Charniak napominju da iako prozodija ne pomaže u točnosti parsera, može biti od pomoći parseru (npr. kraće vrijeme parsanja).

**3.4.10.1. <NP> višeznačnosti**

U rječniku postoje tri riječi koje u ovom slučaju stvaraju <NP> višeznačnost. Od toga je jedna veznik (ali,C), a dvije su imenice: jedna opća (ala, N+c) i jedna vlastita (Ali,N+vl).

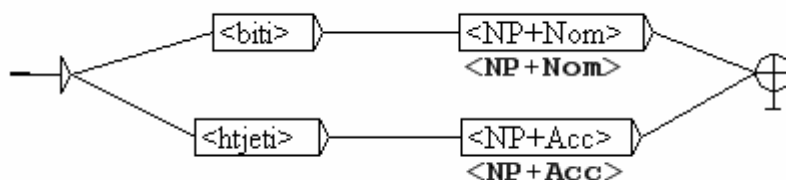
Kako su u dativu i lokativu jednine, oblici za riječ 'ala' identični i glase 'ali', te za riječ 'Ali' (muško ime) nominativ i vokativ jednine također 'Ali' dolazi do višeznačnog obilježavanja tog pojma i kao veznika i kao opće i kao vlastite imenice. U gramatici **nepromjenjive.nog** dodala sam podgrafove kojima sam djelomično razriješila ovu višeznačnost.

Na žalost, nije otklonjena višeznačnost pojma 'ali' kad se on nalazi na početku rečenice i pisan je velikim slovom tako da je taj pojam na toj poziciji označen i kao veznik, i kao opća i kao vlastita imenica. Uvažavam i primjedbu<sup>100</sup> Vjere Lopine koja napominje da se oblik 'ali' može naći kao imenica i iza zareza, kao što je vidljivo i iz primjera kojeg je navela:

✂ Zmaju dajemo blago, ali dajemo hranu.

No, unatoč tome, mišljenja sam da je gramatika ipak opravdana i može ostati kao što je ovdje opisana zbog veoma niske frekvencije pojavljivanja riječi 'ali' kao veznika iza zareza u odnosu na vrlo malu vjerojatnost njezinog pojavljivanja u funkciji imenice.

U mnogo slučajeva se iza pomoćnih glagola <biti> i <htjeti> nalaze imenske skupine označene i kao nominativni skup i kao akuzativni skup pa sam dodatnom gramatikom to pokušala razriješiti na sljedeći način (vidi Sliku 51).

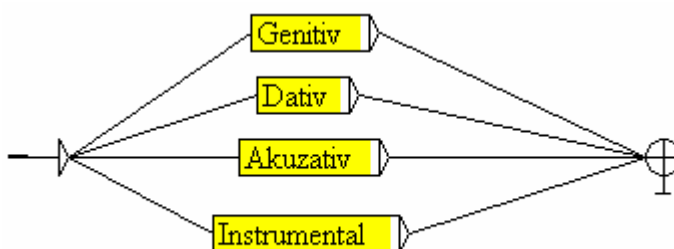


Slika 51: <biti> + nominativna konstrukcija, <htjeti> + akuzativna konstrukcija

<sup>100</sup> Osobna korespondencija.

Znači, ako se iza bilo kojeg oblika glagola <biti> pojavi imenska skupina označena kao nominativna NP skupina, tada ona ostaje samo u nominativu bez obzira na njezine ostale oznake (tj. ako je skupina istovremeno označena i kao nominativna i kao akuzativna). Na isti način, ako se iza bilo kojeg oblika glagola <htjeti> pojavi imenska skupina označena kao akuzativna NP skupina, tada ona zadržava samo tu oznaku (iako istovremeno može biti označena i kao nominativna NP skupina). Ova gramatika ima nisku vrijednost pojavljivanja tj. primjenjuje se skoro na samom kraju analize, tek nakon što su sve ostale mogućnosti za uklanjanje višeznačnosti primijenjene.

Nakon primjene gramatike VP\_NP\_PP.nog, ostalo je dosta višeznačno označenih NP-ova u neposrednoj blizini s glagolom. Kako je pred kraj izrade ovog rada, bio završen i doktorski rad Nives Mikelić Preradović (Mikelić Preradović, 2008.) primijenila sam valencijske okvire glagola koje je ona popisala i opisala i pretočila ih u sljedeću gramatiku (vidi Sliku 52).



Slika 52: Gramatika VP\_DCobl\_PCobl

Naime, postoje glagoli u hrvatskome jeziku koji imaju obaveznu NP dopunu u genitivu, dativu, akuzativu ili instrumentalu i obaveznu PP dopunu u genitivu, dativu, akuzativu, lokativu ili instrumentalu. To je valencijsko svojstvo posebno navedeno uz svaki glagol u leksikonu (kao što je detaljno opisano u poglavlju 3.2.3.) gdje je NP obavezna dopuna opisana kao <DCobl=jedan od padeža>, a PP dopuna kao <PCobl=jedan od padeža>. Te sam oznake iskoristila i u ovoj gramatici za uklanjanje višeznačno označenih NP-ova i PP-ova koji se u tekstu pojavljuju odmah nakon VP

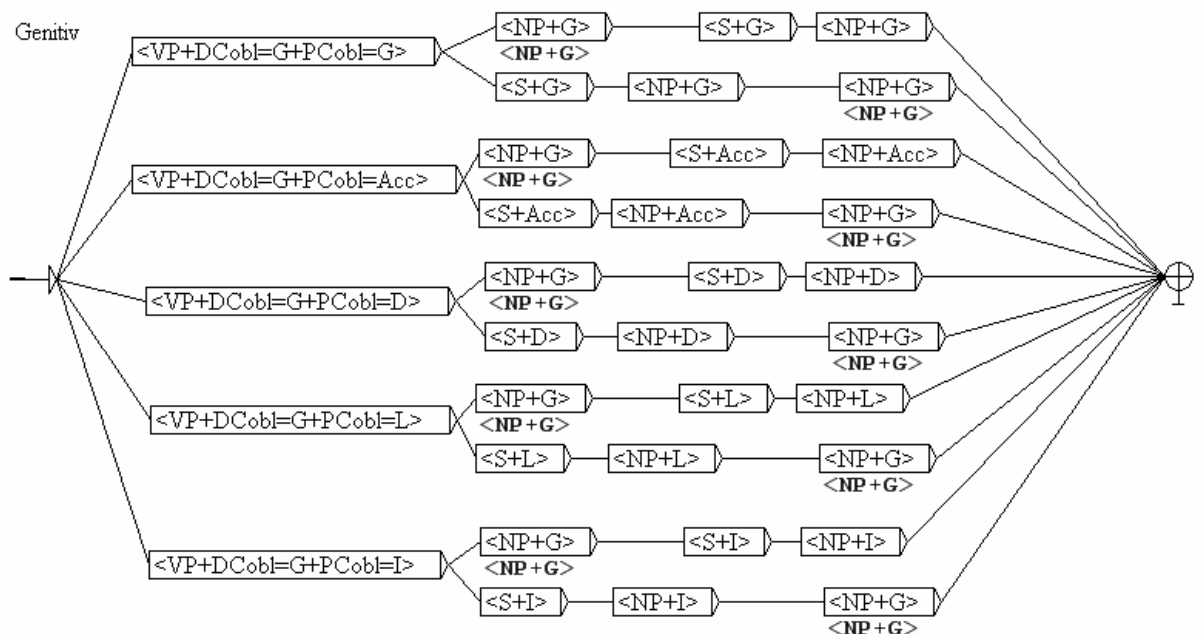
oznake. Slučajeve u kojima se obavezna dopuna ne nalazi u neposrednoj blizini glagola već je dislocirana, za sada nisam obrađivala.

Glavni je graf podijeljen na 4 podgrafova (vidi Sliku 52). Svaki od podgrafova razlikuje se samo u opisu DCobl oznake unutar VP čvora. Usporede li se slike (Slika 53) i (Slika 54) može se primijetiti da su one posve identične uz iznimku svojstva DCobl oznake koja je na prvoj slici u genitivu a na drugoj u dativu.

Ovisno o opisu PCobl oznake unutar VP čvora, NP dopuna i PP dopuna koje slijede mogu se za prvu sliku pojaviti kao <NP=genitiv> + <PP=genitiv> ili u inverziji kao <PP=genitiv> + <NP=genitiv>. Jedan od primjera iz testnog uzorka je glagol 'postaviti' koji u leksikonu ima oznake DCobl=Acc i PCobl=Acc pa je u rečenici

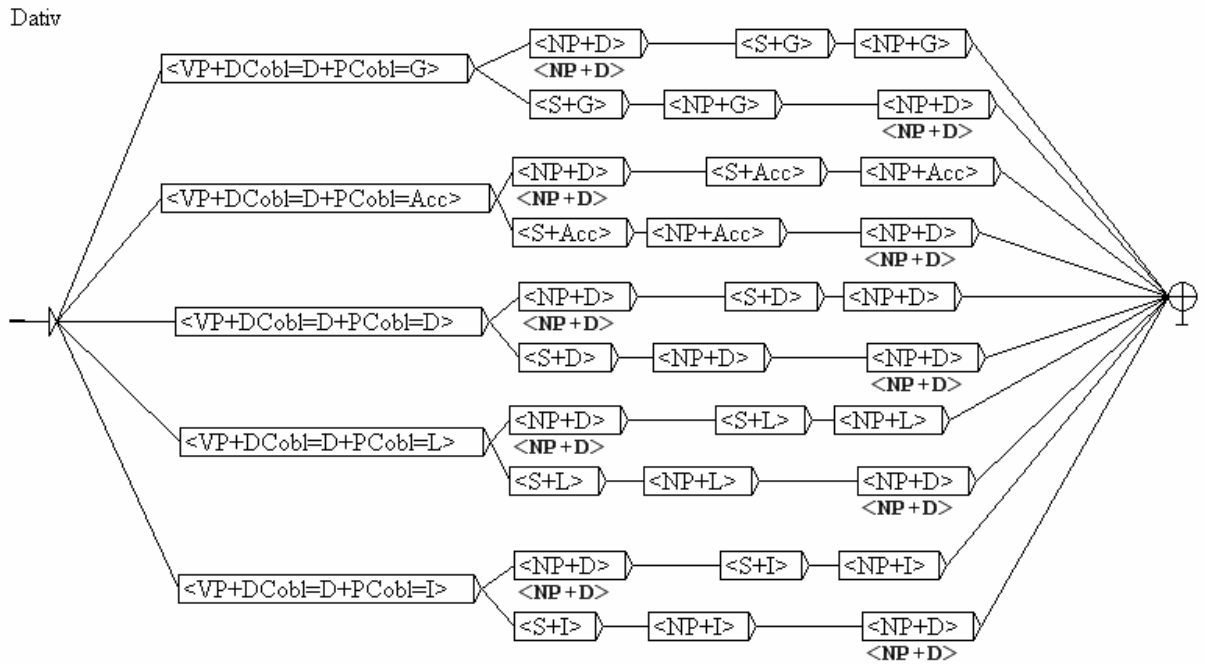
✂ ... postaviti kriterije za ulazak ...

riječ 'kriterije' jednoznačno označeno kao <NP+Acc> a 'za ulazak' kao <PP+Acc>.



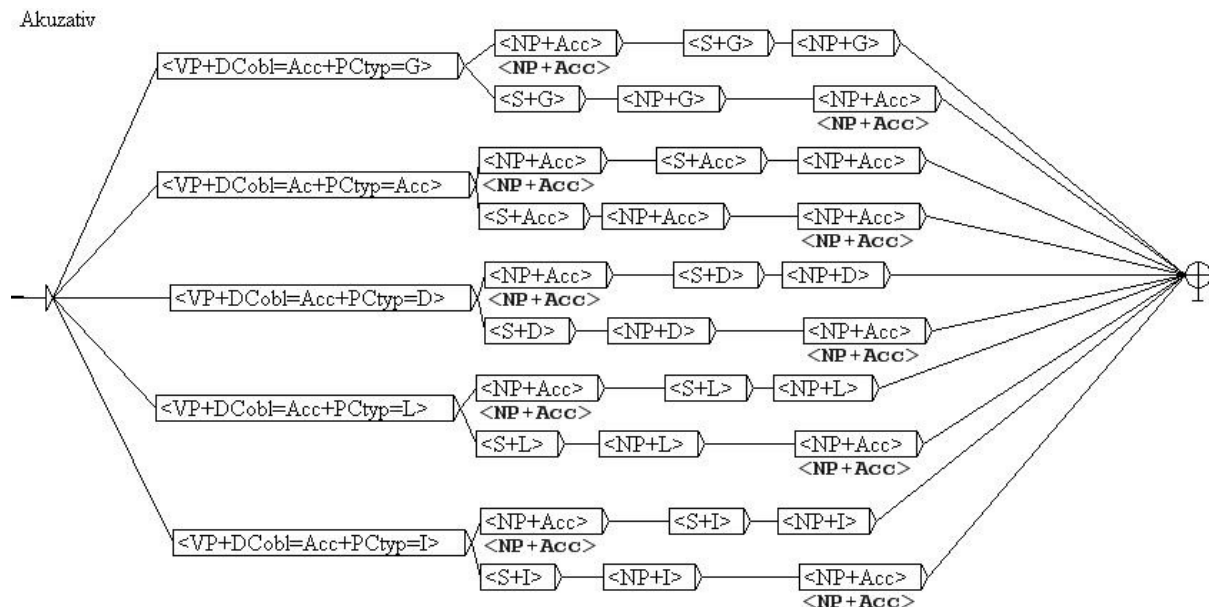
Slika 53: VP sintagma ima obaveznu NP dopunu u genitivu





Slika 54: VP sintagma ima obaveznu NP dopunu u dativu

Nakon kombinirane provjere obaveznih imenskih i prijedložnih dopuna glagolu, slijedi gramatika za prepoznavanje obaveznih imenskih i neobaveznih prijedložnih dopuna.



Slika 55: VP sintagma s obaveznom dopunom u akuzativu i izbornom dopunom u nekom padežu

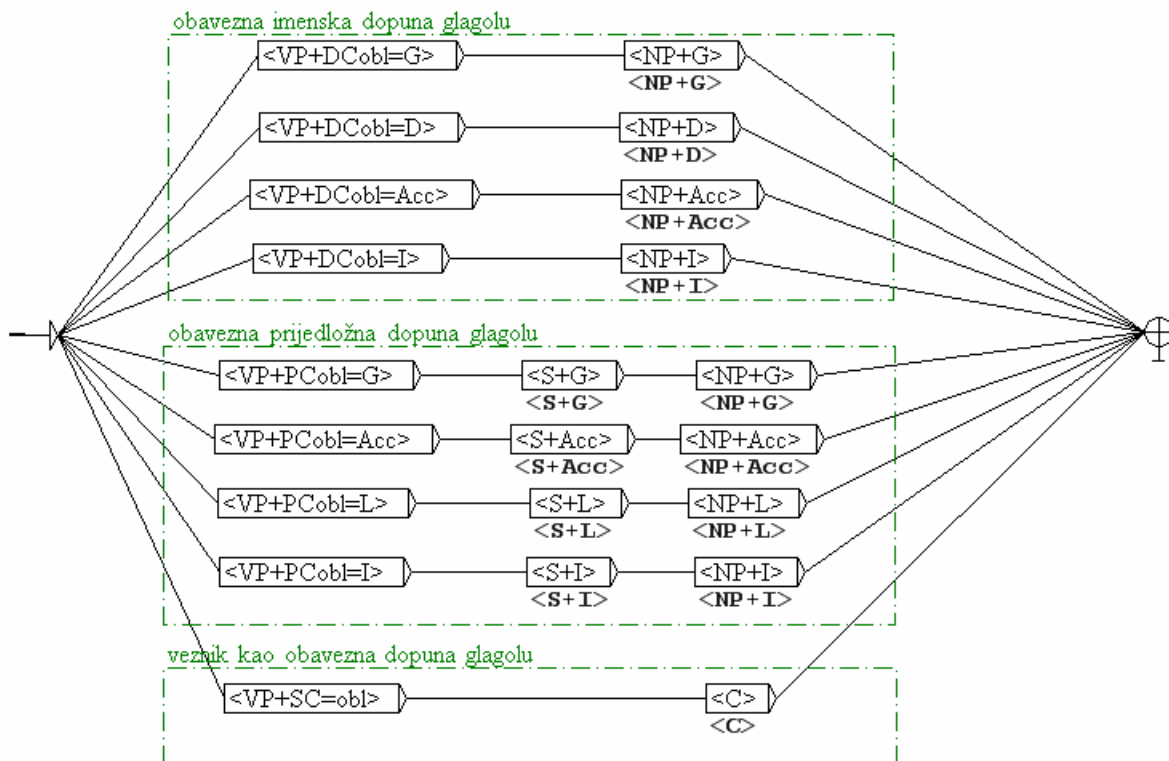
Gramatika je, uz malu zamjenu PCobl atributa s PCtyp atributom, identična upravo opisanoj gramatici. Ovdje navodim samo primjer kada je obavezna padežna dopuna u akuzativu (vidi Sliku 55) dok neobavezna prijedložna dopuna može biti u genitivu, akuzativu, dativu ili lokativu. Poredak obavezne i neobavezne dopune može biti i u inverziji.

Primjeri konkordancije za ovu gramatiku su:

✂ ...će se brzo naći <nacrt zakona> <o poticanju ulaganja>...

✂ ...povuče <svoju odluku> <o vraćanju mandata>...

Gramatika koja potom slijedi provjerava one glagole koji imaju samo jednu od dvije dopune (DCobl ili PCobl) ili pak glagole koji imaju veznik kao obaveznu dopunu (vidi Sliku 56).



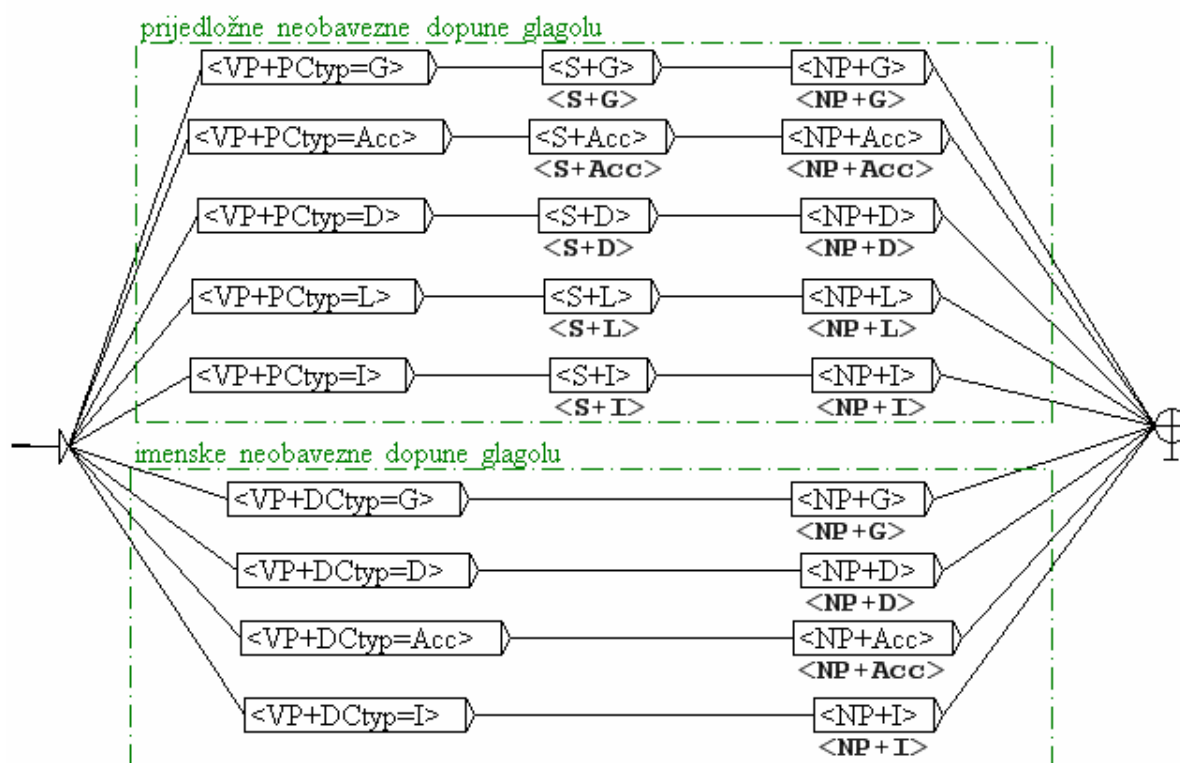
Slika 56: <NP>, <PP> ili <C> kao pojedinačne obavezne dopune glagolu

Primjeri iz teksta za ovu gramatiku su:

njegova posla dok je Račan	kazao da/<C>	su se još tijekom predizborne
suzdržanih ), novi predsjednik Vijeća Škrabalo	kazao je da/<C>	se takvom odlukom jako otežao
granice EU-a. Razgovaralo se,	kaže Rim/<NP+Acc>	, i o primjerenu korištenju jadran:
predsjednik ima mogućnost određivati i	kontrolirati njihov rad/<NP+Acc>	. To se odnosi i na
za ulazak u stranku. Svatko	može potpisati našu pristupnicu/<NP+Acc>	, ali ne može baš svatko
sada prognani Hrvati iz BiH	naći alternativni smještaj/<NP+Acc>	. Riječ je o izuzetno povoljnim
Zlatka Tomčića, Mesić i Račan	našli su se na ručku/<S+L><NP+L>	gdje su dogovorili da će
Bruzellesu konačno priznalo da se	ne mogu čekati nikakve dramatične beogradske promjene/<NP+Acc>	, bez obzira na to što
mandat na raspolaganje, istaknuvši da	ne osjeća krivnju/<NP+Acc>	nitni sram za ono što

Slika 57: Konkordancija za pojedinačne obavezne dopune glagolu

Preostalo je još pronaći i neobavezne pojedinačne <NP> i <PP> dopune glagolu (vidi Sliku 58).



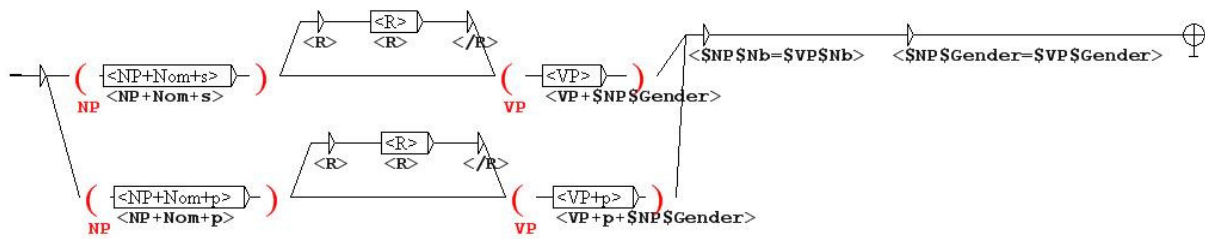
Slika 58: <NP> ili <PP> kao pojedinačne neobavezne dopune glagolu

Neki od primjera iz testnog uzorka su:

osnivačkoj skupštini DC-a Granić izabran za predsjednika/<S+G><NP+G> , a Vesna Škare-Ožbolt za  
želimo biti elitistička stranka, niti se smatramo puritancima/<NP+I> koji u životu nisu ništa  
fond, a s hrvatske se puca vatrenim oružjem/<NP+I> , što Rim smatra pretjeranim. Knjiga  
puca vatrenim oružjem, što Rim smatra pretjeranim/<NP+I> . Knjiga " Kriminal u hrvatskoj pretvorbi

Slika 59: Konkordancija za pojedinačne neobavezne dopune glagolu

Gramatikom Subjekt\_predikat.nog (vidi Sliku 60) opisano je slaganje subjekta i predikata u rečenici između kojih se može pojaviti još i jedan prilog. Subjektni skup, tj. imenska sintagma u nominativu, mora se slagati s glagolskom sintagmom u broju i rodu.



Slika 60: Subjekt\_predikat.nog

Na ovaj su način jednoznačno označene one <NP> sintagme koje su uz nominativ, bile označene i kao neki drugi padež (uglavnom akuzativ). Neki od primjera<sup>101</sup> koji zadovoljavaju ovu gramatiku su:

✂ Pakt konačno prohodao

✂ Zagreb nikada nije otvoreno moljakao

✂ Bruxelles označava

✂ Demokracija se isplati

✂ Konferencija je polučila

✂ Predsjednik je odgovorio

<sup>101</sup> Subjekt je podcrtan jednostruko, predikat dvostruko, a prilog s valovitom crtom.

- ✎ Goran Granić tada je kazao
- ✎ Neki očito nisu shvatili
- ✎ On je istakao
- ✎ Zakon bi nudio
- ✎ Talijansko-hrvatska prekojadranska suradnja bila je
- ✎ Razgovor je poslužio
- ✎ Ta uvijena formulacija odnosi
- ✎ Glavni krijumčarski smjerovi preselili

#### 3.4.10.2. <AP> i <AT> višeznačnosti

U slučajevima kada se atribut i imenica u <AT> sintagmi, te apozicija i imenica u <AP> sintagmi nalaze u genitivu, tada može doći do višeznačnog označivanja sintagmi kao <AT> i kao <AP>. Neki od primjera koji su tako označeni su:

- ✎ *do korištenja državnih nekretnina*
- ✎ *do prihvaćanja novog Zakona*
- ✎ *imenovanja šefova*
- ✎ *iz krugova hercegovačkih Hrvata*
- ✎ *između predsjednika države*
- ✎ *nakon smjene Kosovca*
- ✎ *od početka ulaganja*

Za sada ova višeznačnost nije riješena.

**3.4.10.3. <VP> višeznačnosti**

Primijetimo u konkordanciji za VP gramatiku pogrešno označene VP-ove za pojam 'da' (od glagola dati):

✂	...<da> tako kažem...
✂	...mislim <da> nema ništa...
✂	...rekao sam <da>...
✂	...jedini ovlašten <da> donosi odluke...
✂	...satima čekalo <da> prođe kraj njegova ...
✂	...koji zahtijeva <da> ima...
✂	...ne bih rekao <da> je to...
✂	...je djelovala okolnost <da> je zbog...
✂	...sugestija <da> ga je ...
✂	...nije teško razabrati <da> spor u vrhu...
✂	Općenito se tvrdi <da> je...
✂	Te <da> se ...

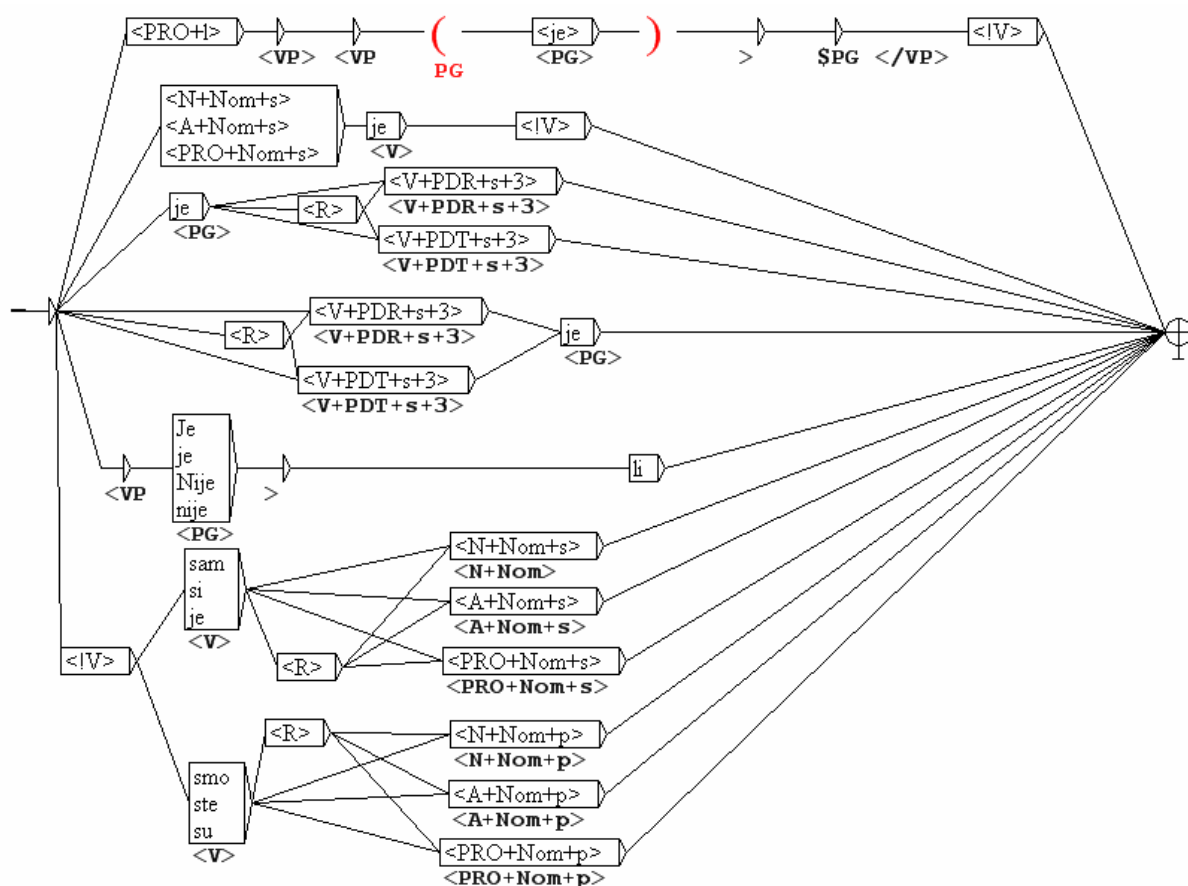
Pogleda li se malo pažljivije, može se uočiti da se u niti jednom slučaju ne radi o prezentu trećeg lica glagola dati, već je riječ o vezniku 'da'. Kako bi se barem djelomično uklonili višeznačni oblici 'da', dodala sam novu gramatiku za prepoznavanje oblika 'da' kao veznika i kao čestice. Ovaj se opis nalazi kao podgraf unutar grafa **nepromjenjive.nog**.

Trebalo je napisati i dodatnu gramatiku za opis riječi 'je'. Naime, glagol 'biti', kao i glagol 'htjeti', se u leksikonu navodi i kao pomoćni glagol, ali i kao glavni glagol pa su i oznake unutar sintaksne analize udvojene.<sup>102</sup> Kako bi se

<sup>102</sup> Dvostruki unos ovih glagola (kao glavnih i kao pomoćnih glagola) u leksikon bio je potreban jer će njihovo međusobno razlikovanje kasnije biti od pomoći kod pisanja gramatika i prepoznavanja

ta višeznačnost, koliko je to moguće, uklonila uvela sam još jednu gramatiku (vidi Sliku 61).

Naime, u slučajevima kada se ispred riječi 'je' nađe zamjenica, pridjev ili imenica u nominativu jednine, a iza nje bilo koja riječ koja nije glagol, tada se 'je' označava kao glavni glagol. Isto je i u slučaju kad se ispred prezenta glagola 'biti' nalazi bilo koja riječ koja nije glagol, a iza nje zamjenica, pridjev ili imenica u nominativu jednine, odnosno množine, tada se taj oblik glagola 'biti' označava kao glavni glagol.



*Slika 61: Gramatika za 'je'*

No, u slučajevima kada se nakon riječi 'je' nalaze glagolski pridjev radni ili glagolski pridjev trpni u trećem licu jednine, s mogućnošću pojavljivanja jednog priloga između njih, tada se 'je' označava kao pomoćni glagol. Isto je i u slučaju inverzije tj. kada se nakon glagolskog pridjeva radnog ili glagolskog pridjeva trpnog u trećem licu jednine ispred kojeg se

složenih glagolskih oblika pogotovo kad se bude radilo o dislociranim glagolskim oblicima, tj. kada se pomoćni i glavni glagol ne nalaze jedan pored drugoga.

može ali ne mora nalaziti još i prilog, pojavi 'je', tada je ono označeno kao pomoćni glagol.

Također, u slučaju da se u tekstu pojave oblici 'je' ili 'nije', na početku rečenice ili u nekom drugom njezinom dijelu, a iza njih riječca 'li', tada se radi o pomoćnom glagolu.

#### 3.4.10.4. <PP> višeznačnosti

Riječ 'pod' se u leksikonu pojavljuje i kao prijedlog koji otvara mjesto akuzativu i instrumentalu, ali i kao imenica u muškom rodu. U gramatici '**nepromjenjive.nog**' pokušala sam jednoznačno odrediti 'pod' kao prijedlog (a ne kao imenicu) u slučaju da je slijedi imenica u akuzativu ili instrumentalu (vidi Sliku 67).

Na taj sam način jednoznačno opisala 'pod' kao prijedlog u primjerima poput:

*biti <pod komandom>*

*biti <pod kontrolom>*

*biti <pod pokroviteljstvom>*

Isti sam problem imala i s pojmom 'oko' koji se u leksikonu također pojavljuje kao prijedlog koji otvara mjesto genitivu, kao imenica u srednjem rodu, te kao prilog uz brojeve kada označava približnu mjeru ili količinu. Zato sam u podgrafu **S** u sklopu gramatike '**nepromjenjive.nog**' riječ 'oko' označila kao prijedlog (a ne kao imenicu) u slučaju da se iza nje nađe imenica, pridjev ili zamjenica u genitivu, a kao prilog ako se iza nje nalazi broj (vidi sliku 67).

Primjeri koji su na ovaj način jednoznačno označeni su:

*<oko> 2500 kuća*

*<oko> ovlasti*



✍ <oko> 20 milijuna njemačkih maraka.

### 3.4.10.5. Nepromjenjive vrste riječi

U grafu za nepromjenjive vrste riječi (prijedlozi, prilozi, veznici, čestice) nalaze se i dodatni opisi za one riječi koje se u tekstu mogu pojaviti i kao promjenjiva vrsta riječi, npr.:

✍ ali – veznik, imenice,

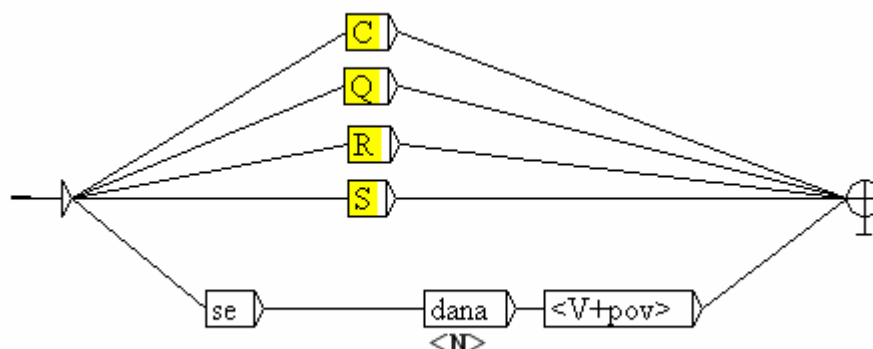
✍ da – veznik, glagol

✍ oko – prijedlog, imenica

✍ pod – prijedlog, imenica

✍ dana – imenica, glagol

✍ kada – imenica, prilog.



Slika 62: Glavni graf za opis nepromjenjivih vrsta riječi

Na glavnom grafu za nepromjenjive vrste riječi (vidi Sliku 62), dodatno je opisana riječ 'dana' kao imenica (a ne glagol) ako se nađe između povratne

zamjenice 'se' i povratnog glagola. Na taj su način uklonjene višeznačne oznake u primjerima poput:

✎ Nekoliko su se **dana** smjenjivale počasne straže...

✎ Ovih se **dana** vratila...

✎ Jednoga će se **dana** iscrpiti...

✎ Ovih se **dana** nalaze...

U podgramatici **C**, koja opisuje veznike, dodatno su dane gramatike za slučajeve kada riječi 'da' i 'ali' trebaju biti označene kao veznici, kao što se vidi u primjerima:

✎ **da** se zaposli...

✎ **da** se osigura...

✎ tvrdio **da** su...

✎ odgovorio **da** je...

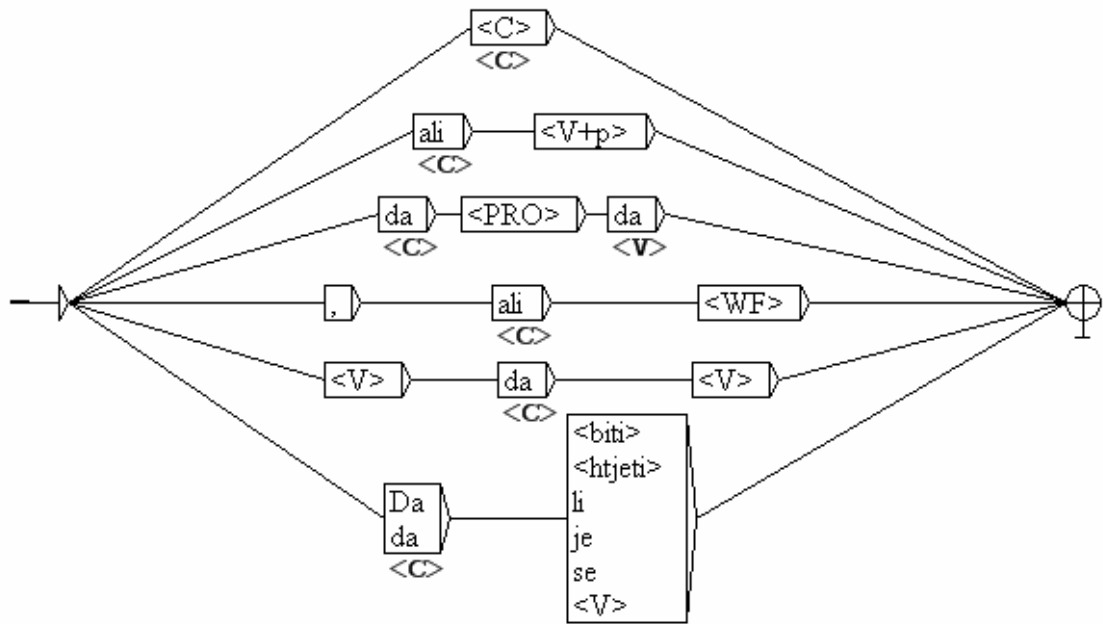
✎ najavio je **da** će sazvati...

✎ , **ali** ipak predstavlja...

✎ , **ali** da šefove...

✎ , **ali** ne može baš...

C

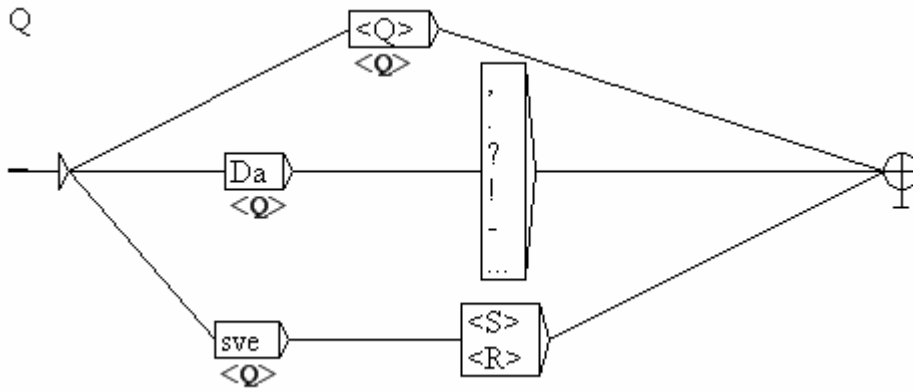


Slika 63: Podgraf za opis veznika

Ova je gramatika uvedena kako bi se otklonile višeznačnosti za riječ 'ali' na način da se ona označi kao veznik, a ne kao imenica, u onim slučajevima kada iza nje slijedi glagol u množini i u slučajevima da je ispred nje zarez a iza nje bilo koja druga vrsta riječi (vidi Sliku 63).

Za riječ 'da' uvedeno je ograničenje da se označi kao veznik (a ne glagol) ako je iza nje zamjenica a potom ponovo riječ 'da', ovaj put označena kao glagol. Isto tako, 'da' treba označiti kao veznik u slučajevima da se pojavi između dva glagola, te u slučaju da se iza nje pojavi bilo koji oblici glagola biti i htjeti, riječca 'li', 'je', 'se' ili neki drugi glagol.

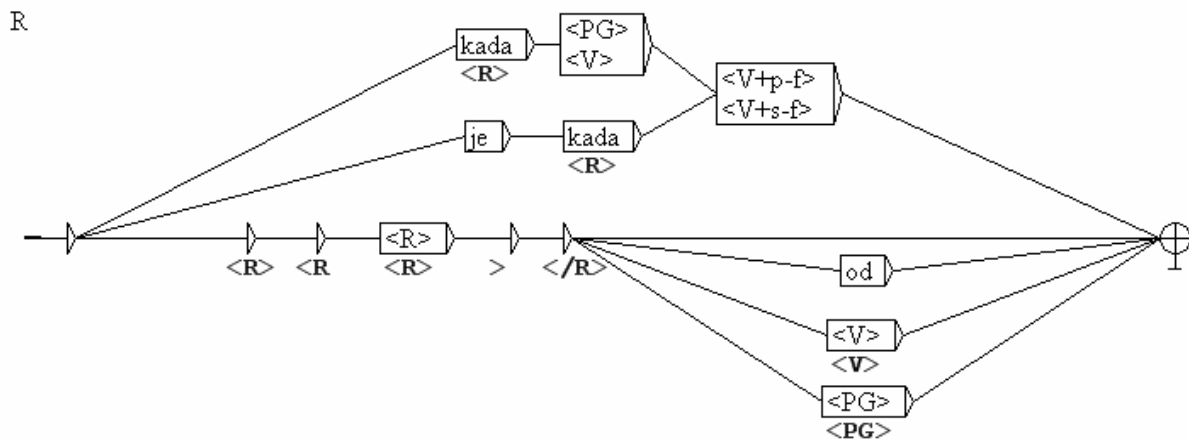
Podgrafom za opis čestica, dodatno je opisan i slučaj kad se riječi 'da' i 'sve' pojavljuju kao čestice.



Slika 64: Podgraf za opis čestica &lt;Q&gt;

Ovim se grafom prepoznaju sve riječi koje su u leksikonu opisane kao čestice, ali i riječ 'da' u slučaju da se nalazi na početku rečenice a odmah iza nje slijedi jedan od interpunkcijskih znakova (, . ? ! - ...). Za riječ 'sve' definirano je da se označi kao čestica (a ne kao zamjenica ili pridjev) u slučaju da je riječ koja je slijedi prilog ili prijedlog (vidi Sliku 64).




Podgraf za opis priloga (vidi Sliku 65) treba ukloniti višeznačno označavanje riječi koje mogu biti i prilog i pridjev. Tu je dodan i opis riječ 'kada' koja je uz prilog bila označena i kao zamjenica i kao imenica.



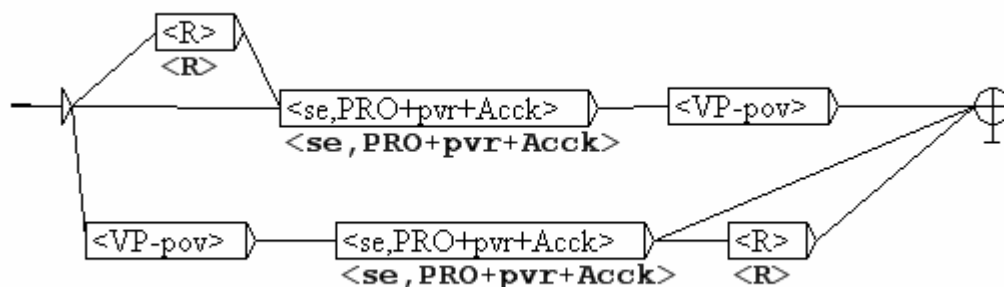
Slika 65: Podgraf za opis priloga &lt;R&gt;

Ovom je gramatikom uklonjeno višeznačno obilježavanje riječi 'kada' u slučajevima kada se iza nje nalazi pomoćni ili glavni glagol iza kojih slijedi još jedan glagol koji je u muškom ili srednjem rodu. Naime, budući riječ

'kada' može nositi i oznaku imenice u ženskom rodu tada, u slučajevima da se iza nje nađe glagol koji nije u ženskom rodu, možemo dosta sigurno pretpostaviti da se ne radi o imenici već o prilogu. O prilogu se radi i u slučaju da se riječ 'kada' nađe između riječi 'je' i glagola koji nije u ženskom rodu. Ostali prilozima se prepoznaju na osnovi leksikona u kojem su navedeni kao prilozima i u slučajevima da se iza njih nađu riječi 'da', pomoćni ili neki glavni glagol.

	<kada> se zbroje...
	<više> od...
	<manje> od ...
	<otvoreno> moljakao ...
	<nedavno> potvrdio ...
	<tragičnije> su stradale...

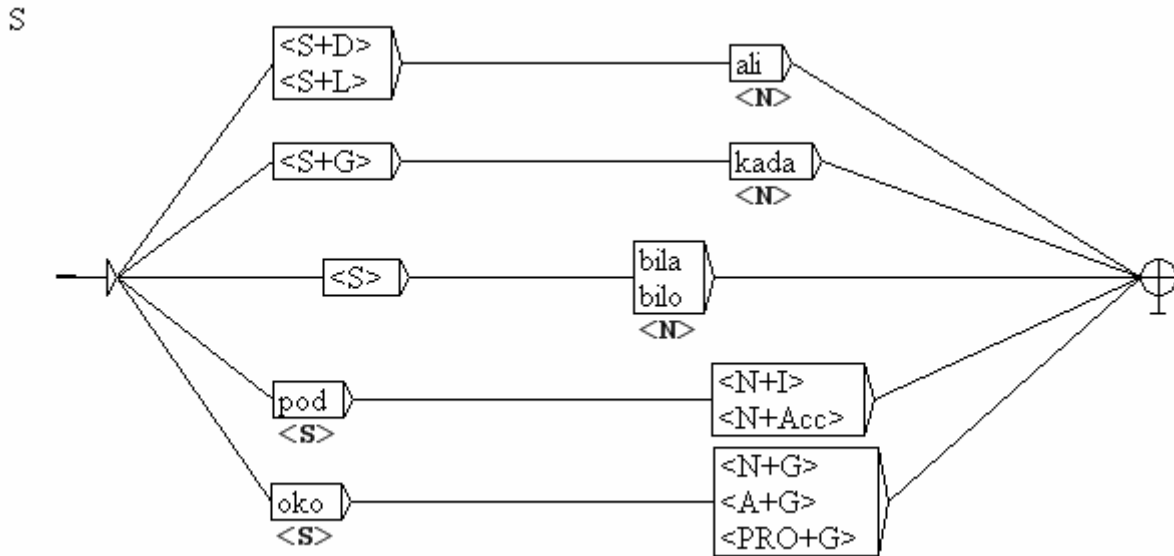
Gramatikom **ObezlicenjeGlagola.nog** (vidi Sliku 66) sam također pokušala detektirati priloge koji se nalaze prije riječi 'se' ako je iza nje glagolska sintagma koja nema oznaku povratnosti i u slučaju da se nađe iza riječi 'se' ako je ispred nje glagolska sintagma koja nema oznaku povratnosti.



Slika 66: ObezlicenjeGlagola.nog

	<više> se ne može govoriti...
	bi se <jednako> odnosio...

Preostao je još podgraf za opis prijedloga (vidi Sliku 67).



Slika 67: Podgraf za opis prijedloga <S>

Ovim se podgrafom jednoznačno označava riječ 'ali' kao imenica u slučaju da se ispred nje nađe prijedlog za dativ ili lokativ, riječ 'kada' kao imenica u slučaju da se ispred nje nađe prijedlog za genitiv i riječi 'bila' i 'bilo' kao imenice u slučaju da je ispred njih bilo koji prijedlog. Bez ove gramatike, riječ 'ali' bi još bila označena i kao veznik, riječ 'kada' i kao prilog, a riječi 'bila' i 'bilo' kao glagoli.

Cijela gramatika **nepromjenjive.nog** primjenjuje se na tekst već kao prva gramatika nakon koje u kaskadnom nizu slijede sve ostale. Razlog tome je što se već na ovom koraku uklanjaju bitne višeznačnosti koje pomažu kod jednoznačnijeg označivanja viših struktura poput <PP>, <VP> i drugih.

### 3.5. Rezultati

Na primjeru od 10 nasumično odabranih rečenica iz već opisanog uzorka teksta, pokazat ću promjene koje su nastale kaskadnom primjenom opisanih gramatika. S obzirom da su gramatike 'nepromjenjive.nog', 'brojevi.nog' i 'glagolJE.nog' vezane uz uklanjanje višeznačnosti na leksičkoj razini, njih ću ovdje preskočiti.

Rečenice su:

- ✎ Ne vidim ozbiljne probleme i sukobe između predsjednika, Vlade, Sabora ili mene.*
- ✎ " Konferencija je polučila uspjeh iznad očekivanja - 2,47 milijardi eura za prvi paket obnove.*
- ✎ U Bavarskoj su socijalni kršćani stvorili supermodernu ekonomiju uz očuvanje tradicionalnih društvenih vrednota.*
- ✎ Dvije velike bankarske krize stajale su Hrvatsku 5,5 milijardi dolara.*
- ✎ Svoju treću knjigu na njemačkom jeziku Jergović će tako predstaviti u Koelnu, Berlinu, Hannoveru i Hamburgu te na lajpciškom sajmu knjiga.*
- ✎ Tu vijest Tomčić je ocijenio kao jedan od prvih koraka do ulaska u NATO i EU.*
- ✎ Vesna Pusić predstavila je novi politički program.*
- ✎ Italija i Hrvatska pripremaju sporazum o strateškoj suradnji.*
- ✎ Koordinacija će svoju prvu sjednicu imati već večeras.*

✍ Time je Granić svoje potencijalne poklonike stavio u dvojbu o ciljevima nove stranke i njezinoj političkoj različitosti u odnosu na ostatak ponude na političkoj sceni.

### 3.5.1. Nakon gramatike *VP\_NP\_PP.nog*:

U primjerima ću crnim slovima ispisati one dijelove koji su jednoznačno označeni prethodnim gramatikama, zelenim slovima dijelove koji su trenutnom gramatikom točno označeni i crvenim slovima one dijelove rečenice koji još nisu jednoznačno označeni.

#### 1. rečenica

```
<VP>Ne vidim</VP>

  <NP TYPE="Acc" Nb="p" Gender="m">

    <NP TYPE="Acc" Nb="p" Gender="m">ozbiljne probleme</NP> i

    <NP TYPE="Acc" Nb="p" Gender="m">sukobe</NP>

  </NP>

  <PP TYPE="G">između

    <NP TYPE="G">

      <NP TYPE="G">predsjednika</NP>,

      <NP TYPE="G">Vlade</NP>,

      <NP TYPE="G">Sabora</NP> ili

      <NP TYPE="G">mene</NP>

    </NP>

  </PP>.
```

U prvoj je rečenici već nakon glavne gramatike sve jednoznačno označeno. Može se činiti kao da 'ozbiljne probleme' ima dvostruke oznake imenske sintagme u akuzativu, no, radi se o koordinaciji dvije imenske



skupine u akuzativu koje su veznikom 'i' povezane u jednu imensku sintagmu, također u akuzativu. Slično je i s pojmovima 'predsjednika', 'Vlade', 'Sabora' i 'mene' koji su povezani veznikom 'ili' u imensku sintagmu u genitivu koja je dio prijedložne sintagme u genitivu.

## 2. rečenica

```

<NP TYPE="G">
  <NP TYPE="Nom" Nb="s" Gender="f">Konferencija</NP></NP>
<VP>je polučila</VP>
  <VP>
    <NP TYPE="Acc">
      <NP TYPE="Nom" Nb="s" Gender="m">uspjeh</NP></NP>
    </VP>
  <PP TYPE="G">iznad
    <NP TYPE="G">očekivanja</NP></PP> -
    <NP TYPE="G">2,47 milijardi eura</NP>
  <PP TYPE="Acc">za
    <NP TYPE="AT" TYPE="Acc">prvi paket obnove</NP></PP>."

```

U drugoj je rečenici višeznačno označen pojam 'Konferencija' kao imenska sintagma u nominativu ili genitivu, te pojam 'uspjeh' kao imenska sintagma u nominativu ili akuzativu, ali i kao glagol.

## 3. rečenica

```

<PP TYPE="L">U
  <NP TYPE="L">Bavarskoj</NP>
  </PP>
<VP>su</VP>
  <NP TYPE="Nom" Nb="p" Gender="m">socijalni kršćani</NP>

```

```

<VP>stvorili</VP>

  <NP TYPE="Acc" Nb="s" Gender="f">supermodernu ekonomiju</NP>

  <PP TYPE="Acc">uz

    <NP TYPE="Acc" Nb="s" Gender="n">očuvanje</NP>

  </PP>

  <NP TYPE="G" Nb="p" Gender="f">tradicionalnih društvenih vrednota</NP>.

```

U trećoj je rečenici već nakon glavne gramatike sve jednoznačno označeno.

#### 4. rečenica

```

  <NP TYPE="Acc" Nb="p" Gender="f">
  <NP TYPE="G" Nb="s" Gender="f">
  <NP TYPE="Nom" Nb="p" Gender="f">Dvije velike bankarske krize</NP></NP></NP>
<VP>stajale su</VP>

  <NP TYPE="Acc" Nb="s" Gender="f">Hrvatsku</NP>

  <NP TYPE="G">5,5 milijardi dolara</NP>.

```

U četvrtoj rečenicu višeznačno je označena grupa 'dvije velike bankarske krize' kao imenska sintagma u nominativu, genitivu ili akuzativu.

#### 5. rečenica

```

  <NP TYPE="Acc" Nb="s" Gender="f">Svoju treću knjigu</NP>

  <PP TYPE="L">na

    <NP TYPE="L">njemačkom jeziku</NP>

  </PP>

  <NP TYPE="Nom" Nb="s">Jergović</NP>

  <VP>će <R>tako</R> predstaviti</VP>

```

```

<PP TYPE="L">u
    <NP TYPE="L">
        <NP TYPE="L">Koelnu</NP>,
        <NP TYPE="L">Berlinu</NP>,
        <NP TYPE="L">Hannoveru</NP> i
        <NP TYPE="L">Hamburgu</NP>
    </NP>
</PP> te
<PP TYPE="L">na
    <NP TYPE="AT" TYPE="L">lajpciškom sajmu knjiga</NP>
</PP>.

```

U petoj je rečenici već nakon glavne gramatike sve jednoznačno označeno. Posebnu pažnju treba skrenuti na koordinaciju imenskih sintagmi u lokativu unutar prijedložne sintagme u lokativu.

## 6. rečenica

```

<NP TYPE="Acc" Nb="s" Gender="f">Tu vijest</NP>
<NP TYPE="Nom" Nb="s">Tomčić</NP>
<VP>je ocijenio</VP> kao <M>jedan</M>
<PP TYPE="G">od
    <NP TYPE="G" Nb="s" Gender="m">prvih koraka</NP>
</PP>
<PP TYPE="G">do
    <NP TYPE="G" Nb="s" Gender="m">ulaska</NP>
</PP>
<PP TYPE="Acc">u
    <NP TYPE="Acc" >
        <NP TYPE="Acc" >NATO</NP> i
        <NP TYPE="Acc" >EU</NP>
    </NP>

```

```

    </NP>

    </PP>.

```

U šestoj je rečenici već nakon glavne gramatike sve jednoznačno označeno.

## 7. rečenica

```

    <NP TYPE="Nom" Nb="s" Gender="f">Vesna Pusić</NP>

    <VP>predstavila je</VP>

    <NP TYPE="Acc" Nb="s" Gender="m">

    <NP TYPE="Nom" Nb="s" Gender="m">novi politički program</NP></NP>.

```

U sedmoj je rečenici višeznačno označena imenska sintagma 'novi politički program' kao nominativni ili akuzativni skup.

## 8. rečenica

```

    <NP TYPE="Nom" Nb="s" Gender="f">

        <NP TYPE="Nom" Nb="s" Gender="f">Italija</NP> i

        <NP TYPE="Nom" Nb="s" Gender="f">Hrvatska</NP>

    </NP>

    <VP>pripremaju</VP>

    <NP TYPE="Acc" Nb="s" Gender="m">

    <NP TYPE="Nom" Nb="s" Gender="m">sporazum</NP></NP>

    <PP TYPE="L">o

        <NP TYPE="L">strateškoj suradnji</NP>

    </PP>.

```

U osmoj je rečenici 'sporazum' označen kao imenska sintagma u nominativu i akuzativu. Ovo je, kao što se vidi i iz prethodnih primjera,

dosta čest slučaj u hrvatskome jeziku i jedan od najvećih problema koji su se pojavili tijekom obrade. Razlog tome je što, osim istih oblika koje mnoge imenice imaju u nominativu i akuzativu jednine, i nominativna i akuzativna imenska sintagma mogu doći ispred ili iza glavnog glagola pa je dodatne provjere trebalo napraviti i u slaganju glagola s imenskom sintagmom u broju i rodu (vidi gramatiku subjekt\_predikat.nog).

## 9. rečenica

```
<NP TYPE="G" Nb="p" Gender="f">
  <NP TYPE="Nom" Nb="s" Gender="f">Koordinacija</NP></NP>
<VP>će</VP>
  <NP TYPE="Acc" Nb="s" Gender="f">svoju prvu sjednicu</NP>
<VP>imati</VP>
<R>već</R> <R>večeras</R>.
```

U devetoj je rečenici imenska sintagma 'koordinacija' označena i kao nominativ i kao genitiv.

## 10. rečenica

```
<NP TYPE="I" Gender="m" Nb="s">Time</NP>
<VP>je</VP>
  <NP TYPE="Nom" Nb="s">Granić</NP>
  <NP TYPE="Acc" Nb="p" Gender="m">svoje potencijalne poklonike</NP>
<VP>stavio</VP>
  <PP TYPE="Acc">u
    <NP TYPE="Acc" Nb="s" Gender="m">dvojbu</NP>
  </PP>
  <PP TYPE="L">o
    <NP TYPE="AT" TYPE="L">ciljevima nove stranke</NP> i
```

```

      <NP TYPE="L">njezinoj političkoj različitosti</NP>

    </PP>

    <PP TYPE="L">u

      <NP TYPE="L">odnosu</NP>

    </PP>

    <PP TYPE="Acc">na

      <NP TYPE="AT" TYPE="Acc" Nb="s" Gender="m">ostatak ponude</NP>

    </PP>

    <PP TYPE="L">na

      <NP TYPE="L">političkoj sceni</NP>

    </PP>.

```

U desetoj je rečenici već nakon glavne gramatike sve jednoznačno označeno.

### 3.5.2. Nakon gramatike *VP\_DCobl\_PCtyp.nog*:

Nakon ove gramatike došlo je do promjena u 8. rečenici.

#### 8. rečenica

```

<NP TYPE="Nom" Nb="s" Gender="f">

  <NP TYPE="Nom" Nb="s" Gender="f">Italija</NP> i

  <NP TYPE="Nom" Nb="s" Gender="f">Hrvatska</NP>

</NP>

<VP>pripremaju</VP>

  <NP TYPE="Acc" Nb="s" Gender="m">sporazum</NP>

  <PP TYPE="L">o

    <NP TYPE="L">strateškoj suradnji</NP></PP>.

```

U osmoj je rečenici '*sporazum*' jednoznačno označen kao imenska sintagma u akuzativu jer glagol '*pripremati*' ima u leksikonu oznaku za obaveznu imensku dopunu u akuzativu i tipičnu (moguću) prijedložnu dopunu u lokativu.

### 3.5.3. Nakon gramatike VP\_DC\_Pcobl.nog

Nakon ove gramatike došlo je do promjena u 2. rečenici i 7. rečenici.

#### 2. rečenica

```
<NP TYPE="G">
    <NP TYPE="Nom" Nb="s" Gender="f">Konferencija</NP></NP>
<VP>je polučila</VP>
    <NP TYPE="Acc">uspjeh</NP>
<PP TYPE="G">iznad
    <NP TYPE="G">očekivanja</NP>
</PP> -
    <NP TYPE="G">2,47 milijardi eura</NP>
<PP TYPE="Acc">za
    <NP TYPE="AT" TYPE="Acc">prvi paket obnove</NP>
</PP>."
```

U drugoj je rečenici '*uspjeh*' sada jednoznačno označen kao imenska sintagma u akuzativu. Razlog tome je što glagol '*polučiti*' u leksikonu ima oznaku za obaveznu imensku dopunu u akuzativu.

**7. rečenica**

```
<NP TYPE="Nom" Nb="s" Gender="f">Vesna Pusić</NP>

<VP>predstavila je</VP>

<NP TYPE="Acc" Nb="s" Gender="m">novi politički program</NP>.
```

U sedmoj je rečenici '*novi politički program*' jednoznačno označeno kao imenska sintagma u akuzativu jer i glagol 'predstaviti' u leksikonu ima oznaku za obaveznu imensku dopunu u akuzativu.

**3.5.4. Nakon gramatike subjekt\_predikat.nog**

Nakon ove gramatike došlo je do promjena u 2. rečenici 4. rečenici i 9. rečenici.

**2. rečenica**

```
<NP TYPE="Nom" Nb="s" Gender="f">Konferencija</NP>

<VP>je polučila</VP>

<NP TYPE="Acc">uspjeh</NP>

<PP TYPE="G">iznad

  <NP TYPE="G">očekivanja</NP></PP> -

  <NP TYPE="G">2,47 milijardi eura</NP>

  <PP TYPE="Acc">za

    <NP TYPE="AT" TYPE="Acc">prvi paket obnove</NP></PP>."
```

U drugoj je rečenici '*konferencija*' jednoznačno označena kao imenska skupina u nominativu jer se kao takva slaže s glagolskom skupinom koja ju slijedi u broju i rodu. Isto je i s rečenicama 4 i 9 koje slijede.



**4. rečenica**

```

<NP TYPE="Nom" Nb="p" Gender="f">Dvije velike bankarske krize</NP>

<VP>stajale su</VP>

<NP TYPE="Acc" Nb="s" Gender="f">Hrvatsku</NP>

<NP TYPE="G">5,5 milijardi dolara</NP>.

```

**9. rečenica**

```

<NP TYPE="Nom" Nb="s" Gender="f">Koordinacija</NP>

<VP>će</VP>

<NP TYPE="Acc" Nb="s" Gender="f">svoju prvu sjednicu</NP>

<VP>imati</VP>

<R>već</R> <R>večeras</R>.

```

**3.5.5. Nakon gramatike *recenica.nog***

Nakon ove gramatike, došlo je do promjena u svim rečenicama na način da su sve dobile oznaku početka <Z> i kraja rečenice </Z>.

**3.5.6. Rezultati za prepoznavanje sintagmi**

Mjerenjem preciznosti, odziva i f-mjere ovdje predloženoga modela razdjelnika dobili su se sljedeći podaci koje u tablici koja slijedi prikazujem paralelno s ručno dobivenim podacima.

103	ZLATNI STANDARD - RDR <sup>104</sup>	ZLATNI STANDARD - MDR <sup>105</sup>			
sintagma	stvaran broj	# <sup>106</sup>	P	O	f
<Z>	135	<b>135</b>	100	100	100
<VP>	439	<b>432</b>	94,44	92,94	93,69
<NP>	843	<b>843</b>	96,56	96,56	96,56
<NP+Nom>	303	364	77,47	93,10	84,56
<NP+G>	204	241	78,42	92,65	84,94
<NP+D>	25	36	55,56	80	65,57
<NP+Acc>	165	298	52,35	94,55	67,39
<NP+Voc>	0	0	/	/	/
<NP+L>	107	98	100	91,59	95,61
<NP+I>	39	48	81,25	100	89,66
<AP>	14	<b>59</b>	22,03	92,86	35,62
<AP+Nom>	7	15	46,67	100	63,64
<AP+G>	1	25	/	/	/
<AP+D>	1	1	/	/	/
<AP+Acc>	2	13	15,39	100	26,67
<AP+Voc>	1	1	100	100	100
<AP+L>	0	1	0	/	/
<AP+I>	2	3	66,67	100	80
<AT>	144	<b>186</b>	74,20	95,83	83,64
<AT+Nom>	49	47	80,85	77,55	79,17
<AT+G>	26	29	79,31	88,46	83,66
<AT+D>	3	7	42,86	100	60
<AT+Acc>	37	69	52,17	97,30	67,93
<AT+Voc>	0	2	0	/	/
<AT+L>	21	21	100	100	100
<AT+I>	8	11	72,73	100	84,21
<PP>	356	<b>334</b>	93,41	87,64	90,43
<PP+G>	87	66	96,97	73,56	83,66
<PP+D>	4	1	100	25	40
<PP+Acc>	92	97	85,57	90,21	87,83
<PP+L>	147	144	95,83	93,88	94,85
<PP+I>	26	26	100	100	100

Iz tablice se jasno vidi da su najbolji rezultati sa 100% vrijednostima za preciznost, odziv i f-mjeru dale gramatike za prepoznavanje apozicije u vokativu, atributa u lokativu i prijedložne sintagme u instrumentalu.

<sup>103</sup> Legenda boja korištenih u tablici

	NAJBOLJI REZULTAT		Vrlo dobar		Loš		Još lošiji
	Odličan		Dobar		Lošiji		Najlošiji rezultat

<sup>104</sup> RDR - Ručno dobiveni rezultati

<sup>105</sup> MDR – Modelom dobiveni rezultati

<sup>106</sup> # - Broj pronađenih sintagmi u tekstu

Uspoređujući općenito sintagme, bez oznaka za padeže, najbolje rezultate ima imenska sintagma čija je f-mjera 96,56%. Nakon toga slijede glagolska sintagma (93,69%), prijedložna sintagme (90,43%), atributska (83,64%) i na poslijetku apozicijska sintagma (35,62%).

### 3.5.7. Rezultati za prepoznavanje padeža <NP> sintagmi

Rezultati za imensku sintagmu bili su znatno lošiji prije primjene dodatnih opisa preuzetih iz CROVALLEX-a. S obzirom da u to vrijeme nisu postojale gramatike za imenski atribut i apoziciju, kao ni gramatike za uklanjanje padežne višeznačnosti za sve NP-ove, a ni višeznačnosti u rodu i broju za NP-ove u nominativu, ovdje dajem samo rezultate za NP sintagmu općenito.

	RDR	PRIJE CROVALLEX-A	POSLIJE CROVALLEX-A
# NP	1001	1099	1088
# višeznačnih NP		437	180 + 14
# točnih jednoznačnih NP		601	746
# pogrešnih jednoznačnih NP		61	39

Tablica 7  
Rezultati za <NP> sintagmu

Dakle, kao što je i vidljivo iz prethodne tablice (Tablica 7), od ukupno 1001 NP-a u testnom tekstu, sustav je pronašao ukupno 1099 prije primjene notacija iz CROVALLEX-a. Od toga ih je 437 bilo višeznačno označeno, što znači da su imali više različitih kombinacija NP oznaka (npr. [NP+D+s] i [NP+Acc+s]), 601 NP je točno jednoznačno označen, dok ih je preostalih 61 označeno pogrešno jednoznačno.

Nakon dodavanja notacija iz CROVALLEX-a, sustav je pronašao ukupno 1 088 NP-ova. Od toga je 746 NP-ova točno jednoznačno označeno a 39 ih je pogrešno jednoznačno označeno. Od jednoznačno pogrešno označenih, svih 39 su ustvari stvarni NP-ovi ali imaju pogrešnu oznaku

padeža, roda ili broja. Broj višeznačno označenih NP-ova je poslije CROVALLEX-a pao na 194 od čega ih je 180 točno označeno kao NP dok ih preostalih 14 pripada nekoj drugoj sintagmi.

Prikazano pomoću standardnih mjera za vrednovanje, situacija je sljedeća:

	PRIJE CROVALLEX-a	POSLIJE CROVALLEX-a
<b>Preciznost</b>	33,31	<b>88,70</b>
<b>Odziv</b>	52,26	<b>96,40</b>
<b>f-mjera</b>	40,69	<b>92,39</b>

Tablica jasno pokazuje koliko su značajni bili podaci o valencijskim okvirima glagola u hrvatskome jeziku u poboljšavanju rezultata lokalnih gramatika za prepoznavanje imenskih sintagmi.

### 3.5.8. Kumulativni rezultati modela

Posljednji u ovom nizu rezultata mjerenja preciznosti, odziva i f-mjere su kumulativni rezultati za ovaj model parsera. Rezultate sam dobila provođenjem sintaksne analize nad 146 jednostavnih rečenica iz korpusa opisanoga u poglavlju 3.1.1.

	ZLATNI STANDARD - RDR	ZLATNI STANDARD - MDR			
sintagma	stvaran broj	#	P	O	f
<Z>	146	<b>146</b>	100	100	100
<VP>	165	<b>164</b>	98,17	97,58	97,87
<NP>	322	<b>360</b>	87,78	98,14	92,67
<NP+Nom>	138	147	93,20	99,28	96,14
<NP+G>	117	122	94,26	98,30	96,23
<NP+D>	8	12	66,67	100	80
<NP+Acc>	102	130	76,15	97,06	85,34
<NP+Voc>	0	0	/	/	/
<NP+L>	69	68	100	98,55	99,27
<NP+I>	26	28	92,86	100	96,30
<AP>	11	<b>20</b>	55	100	70,97
<AP+Nom>	5	7	71,43	100	83,33
<AP+G>	2	4	50	100	66,67
<AP+D>	0	0	/	/	/

<AP+Acc>	1	5	20	100	33,33
<AP+Voc>	0	0	/	/	/
<AP+L>	1	2	50	100	66,67
<AP+I>	2	2	100	100	100
<AT>	49	<b>56</b>	87,50	100	93,33
<AT+Nom>	17	18	94,44	100	97,14
<AT+G>	3	4	75	100	85,71
<AT+D>	2	2	100	100	100
<AT+Acc>	22	28	78,57	100	88
<AT+Voc>	1	1	100	100	100
<AT+L>	18	18	100	100	100
<AT+I>	4	4	100	100	100
<PP>	187	<b>186</b>	99,46	98,93	99,20
<PP+G>	49	50	98	100	98,99
<PP+D>	1	1	100	100	100
<PP+Acc>	45	43	100	95,56	97,73
<PP+L>	73	73	100	100	100
<PP+I>	19	19	100	100	100

Iz pojedinačno prikazanih vrijednosti za preciznost, odziv i f-mjeru, dobila sam i ukupne vrijednosti koje se mogu prikazati tablično:

	Preciznost	Odziv	f-mjera
Model parsera za hrvatski jezik	87,99	99,11	92,34

Vjerujem da su rezultati za prvi pokušaj izrade modela parsera za sintaksnu obradu hrvatskoga jezika izvrsni. Za usporedbu vidi rezultate drugih sličnih modela čiji su rezultati prikazani na str. 43.

## 4. ZAKLJUČAK I PERSPEKTIVE

Ovaj sam rad, nakon uvodnoga dijela, podijelila na dvije zasebne cjeline. U prvoj sam se cjelini osvrnula na teorijsku podlogu praktičnim modelima koje sam detaljno opisala u drugoj cjelini. Prvim sam dijelom rada dala povijesni pregled potreban kako bi se danas uopće moglo govoriti o tehnologiji parsanja. Iz niza gramatika i teoretskih pristupa izradi sintaksnog modela za obradu jezika, odlučila sam se za gramatike ovisnosti i lokalne gramatike koje su mi se svojim teoretskim postavkama činile pogodne za opis flektivno bogatoga i redom riječi slobodnijega jezika kao što je hrvatski. Na kratko sam se osvrnula i na opis predobradnih procedura s naglaskom na razdjelnik kojeg sam ugradila u ovaj prvi model parsera za hrvatski jezik.

Povijest parsera, postojeće definicije, koje se nerijetko razlikuju od jednog autora do drugoga, potom opis zadataka, karakteristika i strukture parsera opisani su neposredno prije opisa triju osnovnih metoda parsanja i njihovih algoritama. Cijeli ovaj teoretski prikaz vodio me je, i navodio, u izradi razdjelnika za hrvatski jezik i izradi modela za sintaksnu analizu hrvatskoga jezika čime je ostvaren zadani cilj ovoga rada.

Imajući na umu različite strukture, metode i prikaze parsanja, jasno se vidi da je model parsera za hrvatski jezik prikazan u ovom radu ***djelomični plitki parser upravljan gramatikom*** čija je analiza prikazana linearno u obliku popisa sintagmi koristeći se ugniježđenim xml zapisom. Svaka sintagma dobivena razdjelnikom ima svoju osnovnu oznaku (NP, VP, PP) ali i sintaksne oznake glave sintagme. Tako npr. radi li se o prijedložnoj sintagmi koja je u lokativu, tada cijela prijedložna sintagma nosi i oznaku lokativa (<PP+L>). Imenska sintagma nosi i oznaku padeža, lica i roda glavne imenice (<NP+L+m+s>). Glagolska sintagma nosi oznaku lica, broja i roda, ali i dodatne sintaksne oznake koje opisuju valencijski okvir glavnog glagola u sintagmi (<VP+3+s+f+DCobl='Acc'+Pctyp='L'>).

Kao prvi model parsera ovoga tipa izrađenoga za hrvatski jezik pokazao se kao iznenađujuće učinkovit, brz i robustan model s visokim stupnjem točnosti i još većim odzivom. Mogli bi smo reći da zadovoljava upravo one kriterije koji se traže od jednog ovakvog sustava koji bi svoju primjenu lako mogao naći u aplikacijama poput pretraživanja i crpljenja obavijesti, crpljenja terminoloških izraza, u leksikografiji, korpusnoj lingvistici, računalnoj lingvistici i NLP-u općenito.

Koraci koji bi se još trebali napraviti u svrhu poboljšanja ovoga modela, počeli su se ocrtavati i izranjati u mojim razmišljanjima pred sam kraj izrade ovoga rada. Model je moguće dodatno poboljšati dodavanjem različitih sintaksnih ali i semantičkih oznaka u leksikon. Te bi oznake omogućile pisanje novih još preciznijih gramatika za prepoznavanje, između ostaloga, i priloga koji su se pokazali kao problematična vrsta riječi u sintaksoj analizi. Vjerujem da će preuzimanje preostalih oznaka iz CROVALLEX-a dodatno pomoći u razrješivanju nekih preostalih višeznačnosti. Lokalne gramatike kojima bi se trebale razriješiti višeznačnosti u rodu i broju među oznakama glagolskih vremena već su u procesu svog nastanka i nadam se da će uskoro biti spremne za testiranje. Dodatno klasificiranje priloga unutar leksikona također bi se trebalo pokazati kao novina kojom ću poboljšati model. Svi ovi koraci trebali bi zajedno rezultirati još preciznijim rezultatima i još većim odzivom modela.

Mišljenja sam da bi dobiveni rezultati bili puno bolji i da se kao ulaz u parser koristi unaprijed označen tekst (Agić, Tadić, 2006, Agić, Tadić, Dovedan, 2008.). No, tu bi tezu tek trebalo isprobati u praksi što ostavljam za, nadam se, neku blisku budućnost.

## 5. RJEČNIK

hrvatski	engleski	kratica
<b>A</b>		
algoritam lijevogkraj	Left-corner algoritam	LC
algoritam maksimalne vjerojatnosti	maximum likelihood algoritam	
algoritam naprijed-nazad	Forward-backward algorithm	
algoritam slijeva	left to right algorithm	LR
analiza funkcija klasa riječi	Word Class Functions Analysis	WCFA
<b>B</b>		
Bayesova metoda	Bayesian method	
Brillova metoda učenja pravila	Brill's rule-learning method	
<b>C</b>		
crpljenje obavijesti	information extraction	
<b>D</b>		
djelomično nenadgledano učenje	partially unsupervised learning	
djelomičnost	partiality	
dubinsko parsanje	deep parsing	
dvopojavnica	bigram	
<b>F</b>		
formalizmi temeljeni na ograničenjima	constraint-based formalisms	
Funkcionalni Generativni Opis	Functional Generative Description	
<b>G</b>		
generalizirani algoritam slijeva	Generalized Left-to-Right Rightmost	
	derivation parser	GLR
generator parsera	parser generator	
gramatika ograničenja	Constraint Grammar	CG
gramatika ovisnosti	dependency grammar	
gramatika rječničkih stabala	Lexicalized Tree Adjoining Grammar	LTAG
gramatika unifikacije ovisnosti	Dependency Unification Grammar	
<b>I</b>		



<i>induktivno ovisno parsanje</i> <i>iskoristivost</i> <i>izraz</i>	<i>Inductive Dependency Parsing</i> <i>utility</i> <i>Phrase</i>	
<b>J</b>		
<i>jednostavni algoritam slijeva</i>	<i>simple LR algorithm</i>	<i>SLR</i>
<i>jednostavni algoritam slijeva s</i> <i>... mogućnošću gledanja unaprijed</i>	<i>Lookahead LR parser</i>	<i>LALR</i>
<b>K</b>		
<i>kaskade konačnih stanja</i> <i>kaskade pretvarača konačnih stanja</i> <i>konekcijske metode</i> <i>koordinacijska višeznačnost</i> <i>razdjelnikom označen korpus</i>	<i>finite state cascades</i> <i>cascade of finite state transducers</i> <i>connectionist methods</i> <i>coordination ambiguity</i> <i>chunked corpus</i>	
<b>L</b>		
<i>Lambekov račun</i>	<i>Lambek calculus</i>	
<b>M</b>		
<i>maksimizacija očekivanja</i> <i>model raspršene aktivacije</i> <i>model rešetke riječi</i> <i>modularna struktura</i>	<i>expectation-maximization</i> <i>spreading activation model</i> <i>word lattice model</i> <i>modular structure</i>	<i>EM</i>
<b>N</b>		
<i>nadgledano strojno učenje</i> <i>naprijed-nazad algoritam</i> <i>navigacijski parser</i> <i>nedeterministički automat</i> <i>nenadgledano strojno učenje</i> <i>neograničeni tekst</i> <i>neusmjerena metoda</i>	<i>supervised machine learning</i> <i>Forward-backward algorithm</i> <i>chart parser</i> <i>non-deterministic automaton</i> <i>unsupervised machine learning</i> <i>unrestricted text</i> <i>non-directional method</i>	<i>NDA</i>
<b>O</b>		
<i>obrada prirodnoga jezika</i> <i>opojavničenje</i> <i>opojavničitelj</i> <i>označena sintagma (razdio)</i> <i>označivač sintagmi</i> <i>označivač temeljen na pravilima</i> <i>označivač temeljen na transformacijama</i> <i>označivanje vrsta riječi</i>	<i>Natural Language Processing</i> <i>tokenization</i> <i>tokenizer</i> <i>chunk</i> <i>phrase marker</i> <i>rule-based tagger</i> <i>transformation-based tagger</i> <i>part-of-speech tagging</i>	<i>NLP</i>       <i>POS tagging</i>
<b>P</b>		
<i>paradigma temeljena na logici</i> <i>parsanje izravno s parsnog stabla</i> <i>parsno stablo</i>	<i>logic-based paradigm</i> <i>parsing directly from trees</i> <i>parse</i>	

<i>plitko parsanje</i>	<i>shallow parsing</i>	
<i>pojavnica</i>	<i>token</i>	
<i>povratno iskoristiv</i>	<i>reusable</i>	
<i>predviđač</i>	<i>predictor</i>	
<i>pretraga prvi najbolji</i>	<i>best-first search</i>	
<i>pretraživanje prvo po dubini</i>	<i>depth-first search</i>	
<i>pretraživanje prvo po širini</i>	<i>breadth-first search</i>	
<i>pretvarač</i>	<i>transducer</i>	
<i>prevoditelj</i>	<i>compiler</i>	
<i>pridodavač</i>	<i>attacher</i>	
<i>pristup «vreća riječi»</i>	<i>bag of words approach</i>	
<i>pristup upravljan gramatikom</i>	<i>grammar-driven approach</i>	
<i>pristup upravljan podacima</i>	<i>data-driven approach</i>	
<i>projekt analize transformacija i diskursa</i>	<i>Transformations and Discourse Analysis Project</i>	TDAP
<i>pronalaženje obavijesti</i>	<i>information retrieval</i>	
<b>R</b>		
<i>razdjelnik</i>	<i>chunker</i>	
<i>razdjeljivanje</i>	<i>chunking</i>	
<i>razobličenje rečenične granice</i>	<i>sentence boundary disambiguation</i>	SBD
<i>ravnoteža</i>	<i>equationality</i>	
<i>razobličenje značenja riječi</i>	<i>word-sense disambiguation</i>	
<i>relacijske informacije</i>	<i>relational information</i>	
<i>rijetko rešetkasta mreža</i>	<i>Sparse Network of Windows</i>	SNoW
<b>S</b>		
<i>samodopunjujuća pretraga</i>	<i>bootstrapping</i>	
<i>silazna metoda</i>	<i>Top-down</i>	
<i>simbol</i>	<i>token</i>	
<i>skener</i>	<i>scanner</i>	
<i>skriveni Markovljev model</i>	<i>Hidden Markov Model</i>	HMM
<i>stablo izvođenja</i>	<i>production tree</i>	
<b>T</b>		
<i>točnost</i>	<i>accuracy</i>	
<b>U</b>		
<i>ulazno-vanjski algoritam</i>	<i>inside-outside algorithm</i>	
<i>unaprijeđene rekurzivne mreže prijelaza</i>	<i>Enhanced Recursive Transition Networks</i>	ERTN
<i>upotpunjivač</i>	<i>completer</i>	
<i>upravljački program</i>	<i>driver</i>	
<i>usmjerena metoda</i>	<i>directional method</i>	
<i>uzlazna metoda</i>	<i>Bottom-up</i>	
<b>V</b>		
<i>višeznačnost pridodavanja</i>	<i>attachment ambiguity</i>	
<b>Z</b>		
<i>zaleđene kombinacije</i>	<i>frozen combinations</i>	

## 6. LITERATURA

<b>A</b>
----------

1. Steven Abney: **Rapid Incremental Parsing with Repair**, u *Proceedings of the 6th New OED Conference: Electronic Text Research*, Waterloo, Ontario, 1-9, 1990.
2. Steven Abney: **Parsing by Chunks**, u *Principle-Based Parsing*, (eds.) R. Berwick, S. Abney, C. Tenny, Kluwer Academic Publishers, 257-278, 1991.
3. Steven Abney: **Prosodic Structure, Performance Structure and Phrase Structure**, u *Proceedings of 5th DARPA Workshop on Speech and Natural Language*, Harriman, New York, 1992.
4. Steven Abney: **Measures and Models for Phrase Recognition**, u *Proceedings, ARPA Human Language Technologies Workshop*, Morgan Kaufmann Publishers, San Mateo, CA. 1993.
5. Steven Abney: **Chunks and Dependencies: Bringing Processing Evidence to Bear on Syntax**, u *Computational Linguistics and the Foundations of Linguistic Theory*, (eds.) J. Cole, G. M. Green, J. L. Morgan, CSLI, 145-164, 1995.
6. Steven Abney: **Partial Parsing via Finite-State Cascades**, u *Workshop on Robust Parsing*, (eds.) J. Carroll, ESSLI'96, 8-15, 1996a.
7. Steven Abney: **Part-of-Speech Tagging and Partial Parsing**, u *Corpus-Based Methods in Language and Speech*, (eds.) K. Church, S. Young, G. Bloothoof, Kluwer Academic Publishers, Dordrecht, 1996b.
8. Steven Abney: **Chunk Stylebook**, rukopis (preuzeto u veljači 2006. s <http://www.vinartus.net/spa/96i.pdf>) , 1996c.
9. Željko Agić, Marko Tadić: **Evaluating Morphosyntactic Tagging of Croatian Texts**, u LREC2006 zbornik, Genova, ELRA, Genova-Pariz, 2006.
10. Željko Agić, Marko Tadić, Zdravko Dovedan: **Improving Part-of-Speech Tagging Accuracy for Croatian by Morphological Analysis**, u *Informatica* 32, 445-451, 2008.
11. Salah Aït-Mokhtar, Jean-Pierre Chanod: **Incremental Finite-State Parsing**, u *Proceedings of ANLP-97*, 72-79, 1997.

12. James Allen: **Natural Language Understanding**, The Benjamin Cummings Publishing Company, Inc., Menlo Park, 1987.
13. James Allen: **Natural Language Understanding**, u *The Handbook of Artificial Intelligence*, Vol. IV., (eds.) Avron Barr, Paul R. Cohen, Edward A. Feigenbaum, Addison-Wesley Publishing, 193-249, 1989.
14. James Allen: **Natural Language Understanding**, 2nd edition, The Benjamin Cummings Publishing Company, Inc., Redwood City, 1995.
15. John Aycock, Nigel Horspool: **Faster Generalized LR Parsing**, u *Proceedings of the 8th International Conference on Compiler Construction*, Springer-Verlag, London, 32-46, 1999.
16. John Aycock, R. Nigel Horspool, Jan Janousek, Borivoj Melichar: **Even faster generalized LR parsing**, u *Proceedings of Acta Informatica 37/9*, 633-651, 2001.
17. John Aycock, R. Nigel Horspool: **Practical Earley Parsing**, The Computer Journal, Vol.45, No.6, British Computer Society, 620-630, 2002.

## B

18. Avron Barr, Edward A. Feigenbaum (eds.): **The Handbook of Artificial Intelligence**, Vol. I., Addison-Wesley Publishing, 1981.
19. Avron Barr, Edward A. Feigenbaum (eds.): **The Handbook of Artificial Intelligence**, Vol. II., Addison-Wesley Publishing, 1982.
20. Barić, Lončarić, Malić, Pavešić, Peti, Zečević, Znika: **Hrvatska gramatika**, Školska knjiga, Zagreb, 2005.
21. Tilman Becker, Owen Rambow: **Parsing Free Word-Order Languages in Polynomial Time**, u *3e Colloque International sur les Grammaires d'Arbres Adjoints (TAG+3)*, Technical Report TALANA, Pariz, 1-5, 1994.
22. Tilman Becker, Aravind K. Joshi, Owen Rambow: **Long-Distance Scrambling and Tree Adjoining Grammars**, u Fifth Conference of the European Chapter of the Association for Computational Linguistics, EACL'91, 21-26, 1991.
23. Kenneth R. Beesley, Lauri Karttunen: **Finite State Morphology**, CSLI Publications, Stanford, 2003.
24. Božo Bekavac: **Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima**, doktorska disertacija, Filozofski fakultet u Zagrebu, Zagreb 2005.

25. Robert C. Berwick, Sandiway Fong: **Principle-Based Parsing: Natural Language Processing for the 1990s**, u Artificial Intelligence at MIT: Expanding Frontiers, (eds.) Patrick H. Winston, Sarah A. Shellard, The MIT Press, Cambridge, 286-325, 1990.
26. Gabriel G. Bes, Lionel Lamadon, François Trouilleux: **Verbal chunk extraction in French using limited resources**, (preuzeto u rujnu 2008. s [http://arxiv.org/PS\\_cache/cs/pdf/0408/0408060v1.pdf](http://arxiv.org/PS_cache/cs/pdf/0408/0408060v1.pdf)), 2004.
27. Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer, Salim Roukos: **Towards History-based Grammars: Using Richer Models for Probabilistic Parsing**, u *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 31-37, 1993.
28. Don Blaheta, Eugene Charniak: **Assigning Function Tags to Parsed Text**, u *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, 234-240, 2000. <http://faculty.knox.edu/dblaheta/research/dpb-naacl00.pdf>
29. Damir Boras: **Teorija i pravila segmentacije teksta na hrvatskom jeziku**, doktorska disertacija, Filozofski fakultet, Zagreb, 1998.
30. Thorsten Brants: **Cascaded Markov Models**, u *Proceedings of the 9th conference on European chapter of the Association for Computational Linguistics*, Bergen, Norway, 118-125, 1999.
31. Eric Brill: **Automatic Grammar Induction and Parsing Free Text: A Transformation-Based Approach**, u *Proceedings of the DARPA Speech and Natural Language Workshop*, 237-242, 1993.
32. Norbert Broker: **Separating Surface Order and Syntactic Relations in a Dependency Grammar**, u *Proceedings of COLING-ACL'98*, 174-180, 1998.
33. Sabine Buchholz, Jorn Veenstra, Walter Daelemans: **Cascaded Grammatical Relation Assignment**, u *Proceedings of EMNLP/VLC-99*, University of Maryland, USA, 239-246, 1999.

## C

34. John Carroll: **Parsing**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 233-248, 2003.
35. Carlo Cecchetto, Gennaro Chierchia: **Reconstruction in Dislocation Constructions and the Syntax/Semantics Interface**, u *The Proceedings of*

- the Seventeenth West Coast Conference on Formal Linguistics, (eds.) K. Shahin, S. Blake, E. Kim, CSLI Publications, 132-146, 1999.
36. Christopher F. Chabris: **Artificial Intelligence and Turbo Pascal**, Dow Jones-Irwin, Homewood, Illinois, 1987.
  37. Eugene Charniak: **Statistical Parsing with a Context-free Grammar and Word Statistics**, u *Proceedings of the 14th National Conference on Artificial Intelligence*, AAAI Press/MIT Press, 598-603, 1997.
  38. Kuang-hua Chen, Hsin-Hsi Chen: **Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation**, u *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, 234-241, 1994.
  39. Noam Chomsky: **Three Models for the Description of Language**, u *Information Theory, IEEE Transactions on*, 2(3):113-124, 1956.
  40. Noam Chomsky: **Aspects of the theory of syntax**, The M.I.T. Press, Cambridge, Massachusetts, 1965.
  41. David Clemenceau: **Finite-State Morphology: Inflections and Derivations in a Single Framework Using Dictionaries and Rules**, u *Finite-State Language Processing*, (eds.) E. Roche, Y. Schabes, The MIT Press, London, 67-98, 1997.
  42. Marie-Hélène Côté: **Quantification Over Individuals and Events and the Syntax-Semantics Interface: The Case of the Existential Constructions**, u *The Proceedings of the Seventeenth West Coast Conference on Formal Linguistics*, (eds.) K. Shahin, S. Blake, E. Kim, CSLI Publications, 147-161, 1999.
  43. Doug Cutting, Julian Kupiec, Jan Pedersen, Penelope Sibun: **A Practical Part-of-Speech Tagger**, u *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, 133-140, 1992. (preuzeto u svibnju 2007., <http://www.ccs.neu.edu/home/futrelle/bionlp/papers/cutting-postagger-anlp92.pdf>).

D

44. Walter Daelemans, Sabine Buchholz, Jorn Veenstra: **Memory-Based Shallow Parsing**, u *Proceedings of CoNLL-99*, Bergen, Norway, 1999.

45. Łukasz Dębowski: **Valence extraction using EM selection and co-occurrence matrices**, (preuzeto u rujnu 2008. s <http://arxiv.org/abs/0711.4475v3>), 2007.
46. Steven J. DeRose: **Grammatical Category Disambiguation by Statistical Optimization**, u *Computational Linguistics*, 14, 31-39, (preuzeto u rujnu 2008. s <http://acl.ldc.upenn.edu/J/J88/J88-1003.pdf>), 1988.
47. Bonnie J. Dorr: **Machine Translation: A Principle-Based Approach**, u *Artificial Intelligence at MIT: Expanding Frontiers*, (eds.) Patrick H. Winston, Sarah A. Shellard, The MIT Press, Cambridge, 326-361, 1990.
48. Bonnie J. Dorr, Lisa Pearl, Rebecca Hwa, Nizar Habash: **DUSTer: A Method for Unraveling Cross-Language Divergences for Statistical Word-Level Alignment**, u *Proceedings of the Fifth Conference of the Association for Machine Translation in the Americas, AMTA-2002*, Tiburon, California, 2002.
49. Zdravko Dovedan, **Formalni jezici: sintaksna analiza**, Zavod za informacijske studije, Zagreb, 2003.

## E

50. Jay Early: **An Efficient Context-Free Parsing Algorithm**, u *Communications of the ACM*, 6(8), 451-455, 1970.
51. Donald Engel, Eugene Charniak, Mark Johnson: **Parsing and Disfluency Placement**, u *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, 49 – 54, 2002.
52. Tomaž Erjavec (ed.): **Specifications and Notation for MULTeXt-East Lexicon Encoding**, MULTeXt-East Report, Concede Edition D1.1F/Concede, Jožef Stefan Institute, Ljubljana, 2001.
53. Tomaž Erjavec: **Multext-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora**, u *Proceedings of the Fourth International Conference on Language Resources and Evaluation, ELRA*, Lisbon-Paris, 1535-1538, 2004.

## G

54. Gregory Grefenstette: **Light Parsing as Finite State Filtering**, u *Extended finite state models of language*, 86-94, 1999.
55. Michelle L. Gregory, Mark Johnson, Eugene Charniak: **Sentence-Internal Prosody Does not Help Parsing the Way Punctuation Does**, (preuzeto u

veljači 2007., [http://acl.ldc.upenn.edu/hlt-naacl2004/main/pdf/50\\_Paper.pdf](http://acl.ldc.upenn.edu/hlt-naacl2004/main/pdf/50_Paper.pdf)).

56. Maurice Gross: **Local Grammars and their representation by finite automata**, u *Data, Description, Discourse: Papers on the English Language in honour of John McH sinclair*, (ed.) M. Hoey, 26-38, 1993.
57. Maurice Gross: **On counting meaningful units in texts**, u JADT, Rim, Italija, 1995a.
58. Maurice Gross: **Representation of finite utterances and the automatic parsing of texts**, u *Language Research* 31, 2, 291-307, 1995b.
59. Maurice Gross: **The Construction of Local Grammars**, u *Finite-State Language Processing*, (eds.) E. Roche, Y. Schabes, MIT Press, London, 329-354, 1997.
60. Maurice Gross: **A bootstrap method for constructing local grammars**, u *Proceedings of the Symposium on Contemporary Mathematics*, (ed.) N. Bokan, 229-250, 1999.
61. Dick Grune: **Parsing – Who Needs It?**, predavanje na Markup Technologies '98, Chicago, <http://www.cs.vu.nl/~dick/mt98.ps>, 1998.
62. Dick Grune, Criel Jacobs: **Parsing Techniques: A Practical Guide**, Ellis Horwood Limited, West Sussex, 1998.

<b>H</b>
----------

63. Udo Hahn, Geert Adriaens: **Parallel Natural Language Processing: Background and Overview**, u *Parallel Natural Language Processing*, ed. G. Adriaens, U. Hahn, Ablex Publishing Corporation, New Jersey, 1-134, 1994.
64. Udo Hahn, Susanne Schacht, Norbert Bröker: **Concurrent, object-oriented natural language parsing: the ParseTalk model**, u *International Journal of Human-Computer Studies*, 41 (1/2), 179-222, (preuzeto u lipnju, 2006.), [http://library.thinkquest.org/18242/data/resources/oo\\_parse.pdf](http://library.thinkquest.org/18242/data/resources/oo_parse.pdf), 1994.
65. Hermann Helbig, Sven Hartrumpf: **Word Class Functions for Syntactic-Semantic Analysis**, u *Proceedings of the 2nd International Conference on Recent Advances in Natural Language Processing (RANLP'97)*, 312-317, Tzigov Chark, 1997.
66. Lynette Hirschman i Inderjeet Mani: **Evaluation**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 414-429, 2003.



67. James E. Hoard: **Language understanding and the emerging alignment of linguistics and natural language processing**, u *Using Computers in Linguistics: A Practical Guide*, (eds) J. Lawler, H. Aristar Dry, Routledge, London, 197-230, 1998.
68. Julia Hockenmaier, Peter Young: **Non-local scrambling: the equivalence of TAG and CCG revisited**, u *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms*, Tübingen, Njemačka, 2008.
69. R. Nigel Horspool, Michael Whitney: **Even Faster LR Parsing**, u *Software – Practice & Experience*, 20(6), 515-535, 1990.
70. Rebecca Hwa: **Sample Selection for Statistical Parsing**, u *Computational Linguistics* 30(3):253-276, 2004.
71. Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, Okan Kolak: **Bootstrapping Parsers via Syntactic Projection across Parallel Texts**, u *Natural Language Engineering* 1, Cambridge University Press, 1-15, 2004.

## I

72. Nancy Ide, Jean Véronis: **Word Sense Disambiguation: The State of the Art**, u *Computational Linguistics*, 24:1, 1-40, (preuzeto u lipnju, 2006.), <http://www.up.univ-mrs.fr/veronis/pdf/1998wsd.pdf>, 1998.
73. Kentaro Inui, Virach Sornlertlamvanich, Hozumi Tanaka, Takenobu Tokunaga: **A New Formalization of Probabilistic GLR Parsing**, u *Proceedings of International Workshop on Parsing Technologies*, 123-134, 1997.

## J

74. Timo Järvinen, Pasi Tapanainen: **Towards an implementable dependency grammar**, u *CoLing-ACL'98 workshop Processing of Dependency-based Grammars*, (eds.) Kahane i Polguère, Montreal, Canada, 1-10, 1998.
75. Karen Jensen, George E. Heidorn: **The Fitted Parse: 100% Parsing Capability in a Syntactic Grammar of English**, u *Proceedings of the Conference on Applied Natural Language Processing, ACL*, 93-98, 1983.
76. Mark Johnson: **Finite-state Approximation of Constraint-based Grammars using Left-corner Grammar Transformations**, u *Proceedings of 36th Annual Meeting of the Association for Computational Linguistics and 17th*

---

*International Conference on Computational Linguistics*, Montreal, Kanada, 619-623, 1998.

77. Bernard E.M. Jones: **Exploring the role of punctuation in parsing natural text**, u *Proceedings of the 15th conference on Computational linguistics – Volume 1*, Kyoto, Japan, 421-425, 1994.
78. Aravind K. Joshi, Phil Hopely: **A Parser from Antiquity**, u *Natural Language Engineering, Volume 2, Issue 4*, 291-294, 1996.
79. Aravind K. Joshi, B. Srinivas: **Disambiguation of Super Pars of Speech (or Supertags): Almost Parsing**, u *Proceedings of the 15th International Conference on Computational Linguistics, COLING'94*, Kyoto, Japan, 154-160, 1994.
80. Daniel Jurafsky, James H. Martin: **Speech and Language Processing: An Introduction to Natural Language Processing**, Computational Linguistics, and Speech Recognition, Prentice Hall, New Jersey, 2000.

<b>K</b>
----------

81. Tuomo Kakkonen: **Robustness Evaluation of Two CCG, a PCFG and a Link Grammar Parsers**, u *Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznan, Poland, 2007.
82. Tuomo Kakkonen: **Framework and Resources for Natural Language Parser Evaluation**, doktorska disertacija, Department of Computer Science and Statistics, University of Joensuu, Finska, 2007.
83. Dae-Hyang Kang: **Scrambling in Universal Grammar: an analysis of scrambling as optional movement in Korean and other languages**, doktorska disertacija na Institut fuer Linguistik/Anglistik der Universitaet Stuttgart, 2005.
84. Ronald M. Kaplan: **Syntax**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 70-90, 2003.
85. Fred Karlsson: **Designing a parser for unrestricted text**, u *Constraint Grammar: A language-independent system for parsing unrestricted text*, (eds.) F. Karlsson, A. Voutilainen, J. Heikkilä, A. Anttila, Mouton de Gruyter, Berlin; New York, 1-40, 1995.

- 
86. Fred Karlsson: **The formalism and environment of Constraint Grammar Parsing**, u *Constraint Grammar: A language-independent system for parsing unrestricted text*, (eds.) F. Karlsson, A. Voutilainen, J. Heikkilä, A. Anttila, Mouton de Gruyter, Berlin; New York, 41-88, 1995.
87. Lauri Karttunen: **Finite-State Technology**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 339-357, 2003.
88. Boris Katz: **Using English for Indexing and Retrieving**, u *Artificial Intelligence at MIT: Expanding Frontiers*, (eds.) Patrick H. Winston, Sarah A. Shellard, The MIT Press, Cambridge, 134-165, 1990.
89. André Kempe, Lauri Karttunen: **Parallel Replacement in Finite State Calculus**, [www. \(preuzeto u rujnu 2006.\)](http://www.ru.nl/~kempe/), 1996.
90. Dan Klein, Christopher D. Manning: **An  $O(n^3)$  Agenda-Based Chart Parser for Arbitrary Probabilistic Context Free Grammars**, (preuzeto u rujnu 2007. [http://www.cs.berkeley.edu/~klein/papers/klein\\_and\\_manning\\_probabilistic\\_chart\\_parser-TR.pdf](http://www.cs.berkeley.edu/~klein/papers/klein_and_manning_probabilistic_chart_parser-TR.pdf)).
91. Kimmo Koskenniemi: **Representations and Finite-State Components in Natural Language**, u *Finite-State Language Processing*, (eds.) E. Roche, Y. Schabes, The MIT Press, London, 99 – 116, 1997.
92. Marcus Kracht: **The Combinatorics of Cases**, u *Research on Language and Computation 1*, Kluwer Academic Publishers, Dordrecht, 59-97, 2003.
93. Miroslav Kržak, Damir Boras: **Rječnička baza hrvatskog književnog jezika**, *Informatologia Yugoslavica* 17, 3-4, Referentni centar Sveučilišta u Zagrebu, Zagreb, 223-242, 1985.
94. Jonas Kuhn: **Corpus-based Learning in Stochastic OT-LFG: Experiments with a Bidirectional Bootstrapping Approach**, eds. M. Butt, T. Holloway King, u *Proceedings of the LFG02 Conference*, National Technical University of Athens, Athens, 239-257, 2002.
95. Jonas Kuhn: **Experiments in Parallel-Text Based Grammar Induction**, (preuzeto u travnju 2007, <http://uts.cc.utexas.edu/~jonask/kuhn-acl04-final.pdf>), 2004.
96. Julian Kupiec: **An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora**, u *Proceedings of the 31st Annual Meeting of the ACL*, 17-22, <http://acl.ldc.upenn.edu/P/P93/P93-1003.pdf>, 1993.

**L**

97. Bong Yeung Tom Lai, Changning Huang: **Dependency Grammar and the Parsing of Chinese Sentences**, u *Proceedings of 1994 Joint Conference of 8th ACLIC and 2nd PacFoCoL*, Kyoto, 63-72, 1994.
98. John M. Lawler, Helen Aristar Dry: **Introduction, u Using Computers in Linguistics: A Practical Guide**, (eds) J. Lawler, H. Aristar Dry, Routledge, London, 1-9, 1998.
99. Steven L. Lytinen: **Integrating syntax and semantics**, u *Machine translation: Theoretical and methodological issues*, ed. S. Nirenburg, Cambridge University Press, London, 302-316, 1987.

**M**

100. David M. Magerman: **Natural language parsing as statistical pattern recognition**, doktorska disertacija, Department of computer science, Stanford University, 1994.
101. Christopher D. Manning, Bob Carpenter: **Probabilistic Parsing Using Left Corner Language Models**, u *Proceedings of the 5th International Workshop on Parsing Technologies*, Boston, USA, 147-158, 1997.
102. Christopher D. Manning, Hinrich Schutze: **Foundation of Statistical Natural Language Processing**, The MIT Press, Cambridge, 2003.
103. Carlos Martin-Vide: **Formal Grammars and Languages**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 157-177, 2003.
104. Oliver Mason: **Automatic Processing of Local Grammar Patterns**, *Proceedings of the 7th Annual CLUK (th UK special-interest group for computational linguistics) Research Colloquium*, (preuzeto u rujnu, 2006. sa <http://www.cs.bham.ac.uk/~mgl/cluk/papers/mason.pdf>), 2004.
105. Ryan McDonald, Joakim Nivre: **Introduction to Data-Driven Dependency Parsing**, skripta za European Summer School in Logic Language and Information, (preuzeto u veljači 2008. sa: <http://dp.esslli07.googlepages.com>), 2007.
106. Philippe McLean, Nigel Horspool: **A Faster Early Parser**, u *Proceedings of Intl. Conference on Compiler Construction (CC'96)*, Linkoping, Sweden, April 1996, LNCS vol. 1060, Springer-Verlag, pp. 281-293. (preuzeto

- u rujnu, 2006. sa <http://webhome.cs.uvic.ca/~nigelh/Publications/fastEarley.pdf>).
107. Beata Megyesi: **Shallow Parsing with POS Taggers and Linguistic Features**, u *Journal of Machine Learning Research* 2, (eds.) J. Hammerton, M. Osborne, S. Armstrong, W. Daelemans, 639-668, 2002.
  108. Igor Mel'čuk: **Dependency in Linguistic Description: Concepts and Problems**, u *Dependency and Valency – An International Handbook of Contemporary Research, vol.1*, Berlin, 188-229, 2003.
  109. Nives Mikelić Preradović: **Pristupi izradi strojnog tezaurusa za hrvatski jezik**, doktorska disertacija, Filozofski fakultet, Zagreb, 2008.
  110. Andrei Mikheev: **Text Segmentation**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 201-218, 2003.
  111. M. Moguš, M. Bratanić, M. Tadić: **Hrvatski čestotni rječnik**, Zavod za lingvistiku i Školska knjiga, Zagreb, 1999.
  112. Mehryar Mohri: **Local Grammar Algorithms**, (preuzeto u srpnju, 2006. sa <http://www.cs.nyu.edu/~mohri/postscript/kos.pdf>), 87-96, 2005.
  113. Mehryar Mohri: **Syntactic Analysis By Local Grammars Automata: An Efficient Algorithm**, (preuzeto u srpnju 2006., [http://arxiv.org/PS\\_cache/cmp-lg/pdf/9407/9407002v1.pdf](http://arxiv.org/PS_cache/cmp-lg/pdf/9407/9407002v1.pdf)), 1994.
  114. Antonio Molina, Ferran Pla: **Shallow Parsing using Specialized HMMs**, u *Journal of Machine Learning Research* 2, (eds.) J. Hammerton, M. Osborne, S. Armstrong, W. Daelemans, 595-613, 2002.
  115. Robert C. Moore: **Improved Left-Corner Chart Parsing for Large Context-Free Grammars**, u *New Developments in Parsing Technology*, (eds.) H. Bunt, J. Carroll, G. Satta, Kluwer Academic Publishers, 185-201, 2004.
  116. Marcia Muñoz, Vasin Punyakanok, Dan Roth, Dav Zimak: **A Learning Approach to Shallow Parsing**, u *Proceedings of EMNL-WVLC'99*, Association for Computational Linguistics, 168-178, 1999.

117. Mark-Jan Nederhof: **Generalized Left-Corner Parsing**, u *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, Nizozemska, 305-314, 1993.

118. Mark-Jan Nederhof, Giorgio Satta: **Tabular Parsing**, u *Formal Languages and Applications, Studies in Fuzziness and Soft Computing* 148, Springer, 529-549, 2004.
119. Mark-Jan Nederhof, Giorgio Satta: **Probabilistic Parsing Strategies**, u *Descriptive Complexity of Formal Systems (DCFS), Pre-Proceedings of a Workshop*, (eds.) J. Dassow, M. Hoeberechts, H. Jurgensen, D. Wotschke, Report No. 586, Department of Computer Science, University of Western Ontario, London, Canada, 216-230, 2002.
120. Peter Neuhaus, Udo Hahn: **Restricted Parallelism in Object-Oriented Lexical Parsing**, u *Proceedings of the 16th conference on Computational linguistics – Volume 1*, Copenhagen, Denmark, 502-507, 1996.
121. Grace Ngai, David Yarowsky: **Rule Writing or Annotation: Cost-efficient Resource Usage for Base Noun Phrase Chunking**, u *Proceedings of the 38th ACL*, 2000.
122. Joakim Nivre: **Dependency Grammar and Dependency Parsing**, MSI Report 05133. <http://www.msi.vxu.se/users/nivre/papers/05133.pdf>, (preuzeto u lipnju, 2006.), 2005.
123. Joakim Nivre: **Inductive Dependency Parsing**, Springer, Dordrecht, Nizozemska, 2006.

## O

124. Kemal Oflazer: **Computational Morphology**, ESSLLI lectures 2006.
125. Miles Osborne: **Shallow Parsing as Part-of-Speech Tagging**, u *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 145-147, 2000.

## P

126. Fernando C.N. Pereira, Stuart M. Shieber: **The semantics of grammar formalisms seen as computer languages**, u *Proceedings of the Tenth International Conference on Computational Linguistics*, Stanford University, Stanford, California, 123-129, 1984.
127. Vladimir Pericliev, Alexander Grigorov: **Parsing a flexible word order language**, u *Proceedings of the 15th conference on Computational linguistics – Volume 1*, Kyoto, Japan, 391-395, 1994.

**R**

128. Lance, A. Ramshaw, Mitchell P. Marcus: **Text Chunking using Transformation-Based Learning**, u *Proceeding of the Third Workshop on Very Large Corpora*, 82-94, 1995.
129. Victor Raskin: **Linguistics and natural language processing**, u *Machine translation: Theoretical and methodological issues*, ed. S. Nirenburg, Cambridge University Press, London, 42-58, 1987.
130. Charles Rich, Richard C. Waters: **The Programmer's Apprentice**, u *Artificial Intelligence at MIT: Expanding Frontiers*, (eds.) Patrick H. Winston, Sarah A. Shellard, The MIT Press, Cambridge, 166-195, 1990.
131. C.J. van Rijsbergen: **Information Retrieval**, Butterworths, 1979.
132. Brian Roark, Mark Johnson: **Efficient probabilistic top-down and left-corner parsing**, u *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Maryland, USA, 421-428, 1999.
133. Emmanuel Roche: **Parsing with Finite-State Transducers**, u *Finite-State Language Processing*, (eds.) E. Roche, Y. Schabes, The MIT Press, London, 241 – 282, 1997.
134. Emmanuel Roche, Yves Schabes: **Deterministic Part-of Speech Tagging with Finite-State Transducers**, u *Finite-State Language Processing*, (eds.) E. Roche, Y. Schabes, The MIT Press, London, 205 – 240, 1997.

**S**

135. Christer Samuelsson: **Statistical Methods**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 358-375, 2003.
136. Erik F. Tjong Kim Sang, Sabine Buchholz: **Introduction to the CoNLL-2000 Shared Task: Chunking**, u *Proceeding of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 127-132, 2000.
137. Erik F. Tjong Kim Sang: **Memory-Based Shallow Parsing**, u *Journal of Machine Learning Research* 2, (eds.) J. Hammerton, M. Osborne, S. Armstrong, W. Daelemans, 553-594, 2002.
138. Erik F. Tjong Kim Sang, Jorn Veenstra: **Representing Text Chunks**, u *Proceedings of EACL'99*, Bergen, Norveška, 173-179, 1999.

- 
139. Virginia Savova, Leonid Peshkin: **Dependency Parsing with Dynamic Bayesian Network**, (preuzeto u kolovozu 2008. s <http://arxiv.org/abs/cs/0703135v1>), 2007.
140. Hinrich Schütze: **Distributional Part-of-Speech Tagging**, (preuzeto u travnju 2007., [http://arxiv.org/PS\\_cache/cmp-lg/pdf/9503/9503009v1.pdf](http://arxiv.org/PS_cache/cmp-lg/pdf/9503/9503009v1.pdf)).
141. Sanja Seljan: **Leksičko-funkcionalna gramatika hrvatskoga jezika: teorijski i praktični modeli**, doktorska disertacija, Filozofski fakultet, Zagreb, 2003.
142. Stuart M. Shieber: **Sentence disambiguation by a shift-reduce parsing technique**, u Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, MIT, Cambridge, Massachusetts, 113-118, 1983.
143. Stuart M. Shieber: **Constraint-Based Grammar Formalisms: Parsing and Type Inference for Natural and Computer Languages**, A Bradford Book, The MIT Press, Cambridge, 1992.
144. Stuart M. Shieber, Yves Schabes, Fernando C.N. Pereira: **Principles and implementation of deductive parsing**, u The Journal of Logic Programming, Elsevier Science Publishing Co., New York, 1-37, 1993.
145. Peh Li Shiuan, Christopher Ting Hian Ann: **A Divide-and-Conquer Strategy for Parsing**, u *Proceedings of the ACL/SIGPARSE 5th International Workshop on Parsing Technologies*, Santa Cruz, USA, 57-66, 2000.
146. Max D. Silberztein: **The Lexical Analysis of Natural Language**, u Finite-State Language Processing, (eds.) E. Roche, Y. Schabes, The MIT Press, London, 175 – 204, 1997.
147. Max D. Silberztein: **NooJ – manual**, (preuzeto u veljači 2007., <http://www.nooj4nlp.net/NooJ%20Manual.pdf>), 2007.
148. Josip Silić, Ivo Pranjković: **Gramatika hrvatskoga jezika za gimnazije i visoka učilišta**, Školska knjiga, Zagreb, 2005.
149. Gary F. Simons: **The nature of linguistic data and the requirements of a computing environment for linguistic research**, u Using Computers in Linguistics: A Practical Guide, (eds) J. Lawler, H. Aristar Dry, Routledge, London, 10-25, 1998.
150. Wojciech Skut, Thorsten Brants: **Chunk Tagger**, u *ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*, Saarbrücken, 1998.



151. Steve L. Small, Rieger Chuck: **Parsing and Comprehending with Word Experts (A Theory and its Realization)**, u *Strategies for Natural Language Processing*, (eds.) Ringle i Lehnert, Lawrence Erlbaum Associates, 89-147, 1982.
152. George W. Smith: **Computers and Human Language**, Oxford University Press, New York, 1991.
153. B. Srinivas, Christine Doran, Seth Kulick: **Heuristics and Parse Ranking**, u *Proceedings of the 4th international workshop on Parsing Technologies*, Prag, Češka, 224-233, 1995.
154. M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, J. Crim: **Bootstrapping Statistical Parsers from Small Datasets**, (preuzetu u lipnju 2007., <http://acl.ldc.upenn.edu/eacl2003/papers/main/p41.pdf>), 2003.
155. M. Steedman, R. Hwa, S. Clark, M. Osborne, A. Sarkar, J. Hockenmaier, P. Ruhlen, S. Baker, J. Crim: **Example Selection for Bootstrapping Statistical Parsers**, (preuzeto u lipnju 2007., <http://www.cs.pitt.edu/~hwa/naacl03.pdf>).
156. Mark Stevenson, Yorick Wilks: **Word-Sense Disambiguation**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 249-266, 2003.

T

157. Marko Tadić: **Računalna obrada morfologije hrvatskoga književnoga jezika**, doktorska disertacija, Filozofski fakultet u Zagrebu, Zagreb, 1994.
158. Marko Tadić: **Računalna obradba hrvatskoga i nacionalni korpus**, u *Suvremena lingvistika* 41-42, 603-612, 1996.
159. Marko Tadić: **Računalna obradba hrvatskih korpusa: povijest, stanje i perspektive**, u *Suvremena lingvistika* 43-44, 387-394, 1997.
160. Marko Tadić: **Information Retrieval Meets Human Language Technology**, u *Quest for Information, CUC2000 Proceedings CD*, CARNet, Zagreb, 2000a.
161. Marko Tadić: **Building the Croatian-English Parallel Corpus**, u *Proceedings of the Second International Conference on Language Resources and Evaluation*, ELRA, Paris-Athens, 523-530, 2000b.

162. Marko Tadić: **Jezične tehnologije i hrvatski jezik**, Exlibris, Zagreb, 2003.
163. Marko Tadić: **The Croatian Lemmatization Server**, u *Southern Journal of Linguistics* 29, 206-217, 2005.
164. Marko Tadić: **Building the Croatian Dependency Treebank: the initial stages**, u *Suvremena lingvistika* 33, 85-92, 2007.
165. Marko Tadić, Božo Bekavac: **Preparation for POS tagging of Croatian using CLaRK System**, u *Proceedings of RANLP2003 Conference*, Borovets, Bugarska, 455-459, 2003.
166. Wesley Tanaka: **PCFG Expectation Maximization**, (preuzetu u svibnju 2008. s <http://wtanaka.com/system/files/pcfg-exp-max.pdf>), 2003.
167. Pasi Tapanainen, Timo Järvinen: **A non-projective dependency parser**, u *Proceedings of the 5th conference on Applied natural language processing*, Washington DC, 64-71, 1997.
168. Stjepko Težak, Stjepan Babić: **Gramatika hrvatskoga jezika: priručnik za osnovno jezično obrazovanje**, Školska knjiga, Zagreb 1996.
169. Maria Todorova: **The Bulgarian frozen and semi-frozen expressions formalized with NooJ**, (preuzeto u svibnju 2008. s <http://nooj.matf.bg.ac.yu/pptpdf/21MariaTodorova-BulFrozenAndSemi-frozenExpressions.pdf>) , 2006.
- 170.

## V

171. Peter Vanroose: **Part-of-Speech Tagging from an information-theoretic point of view**, u *Proceedings of the 22nd Symposium on Information Theory*, Benelux, 9-14, 2001.
172. Atro Voutilainen: **Three studies of grammar-based surface parsing of unrestricted English text**, u *Publications of the Department of General Linguistics*, University of Helsinki, No.24. 1994.
173. Atro Voutilainen: **NPtool, a detector of English noun phrases**, u *Proceedings of the Workshop on Very Large Corpora*, 48-57, 1993 (preuzeto u veljači 2007., <http://www.aclweb.org/anthology-new/W/W93/W93-0306.pdf>).

174. Atro Voutilainen: **Designing a (Finite-State) Parsing Grammar**, u *Finite-State Language Processing*, (eds.) E. Roche, Y. Schabes, The MIT Press, London, 283 – 310, 1997.
175. Atro Voutilainen, Juha Heikkilä: **Compiling and testing the lexicon**, u *Constraint Grammar: A language-independent system for parsing unrestricted text*, (eds.) F. Karlsson, A. Voutilainen, J. Heikkilä, A. Anttila, Mouton de Gruyter, Berlin; New York, 89-102, 1995.
176. Atro Voutilainen, Timo Järvinen: **Specifying a shallow grammatical representation for parsing purposes**, u *Proceeding of the Seventh Conference on the European Chapter of the Association for Computational Linguistics*, Dublin, 210-214, 1995.
177. Atro Voutilainen: **Part-of-Speech Tagging**, u *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford University Press, Oxford, 219-232, 2003.
178. Atro Voutilainen: **A syntax-based part-of-speech analyser**, u *Proceedings of the 7th conference on European chapter of the Association for Computational Linguistics*, Dublin, Irska, 157-164, (preuzeto u veljači 2007., [http://arxiv.org/PS\\_cache/cmp-lg/pdf/9502/9502012v2.pdf](http://arxiv.org/PS_cache/cmp-lg/pdf/9502/9502012v2.pdf)), 1995.
179. Atro Voutilainen, Lluís Padro: **Developing a hybrid NP parser**, u *Proceedings of the fifth conference on Applied natural language processing*, Washington DC, 80-87, 1997.
180. Kristina Vučković: **Padežne gramatike i razumijevanje hrvatskoga jezika**, magistarski rad, Filozofski fakultet, Zagreb, 2004.
181. Kristina Vučković, Marko Tadić, Zdravko Dovedan: **Rule Based Chunker for Croatian**, u *Proceeding of the Sixth International Conference on Language Resources and Evaluation LREC 2008*, Marakeš: ELRA, 2008.
182. Kristina Vučković, Nives Mikelić Preradović, Zdravko Dovedan: **Verb Valency Enhanced Croatian Lexicon**, u *Proceedings of NooJ 2008*, Budimpešta, Mađarska, 2008.

Z

183. GuoDong Zhou, Jian Su, TongGuan Tey: **Hybrid Text Chunking**, u *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 163-165, 2000.

Ž

184. Tomislava Žubrinić: **Mogućnosti strojnoga označavanja i lematiziranja korpusa tekstova hrvatskoga jezika**, magistarski rad, Filozofski fakultete, Zagreb, 1995.

X
---

185. Chenhai Xi, Rebecca Hwa: **A Backoff Model for Bootstrapping Resources for Non-English Languages**, (preuzeto u lipnju 2007., <http://www.cs.pitt.edu/~hwa/emnlp05.pdf>).

Y
---

186. Akinori Yonezawa, Ichiro Ohsawa: **Object-Oriented Parallel Parsing for Context-Free Grammars**, u *Parallel Natural Language Processing*, ed. G. Adriaens, U. Hahn, Ablex Publishing Corporation, New Jersey, 188-210, 1994.

## **PRILOZI**

## Prilog 1 – Dio zlatnog standarda za testiranje razdjelnika (uzorak2.not)

*U pozadini tog sukoba ipak je bio sukob oko ovlasti predsjednika države te u vezi s nadzorom rada tajnih službu.*

*Prema sadašnjem zakonu predsjednik države je zapovjednik vojske, sukreator je vanjske politike a on također imenuje i šefa Ureda za nacionalnu sigurnost, službe koja nadzire rad obavještajnih službi.*

*Nakon što su preko medija izmijenili optužbe, uz posredništvo predsjednika Sabora Zlatka Tomčića, Mesić i Račan našli su se na ručku gdje su dogovorili da će ubuduće probleme rješavati razgovorom.*

*Nakon tog susreta njih dvojica održali su odvojene konferencije za novinare na kojima su još jednom ponovili kako među njima nema sukoba.*

*Predsjednik države izjavio je da nema sukoba između njega, predsjednika Vlade i Sabora, a ručak i razgovor s Račanom i Tomčićem ocijenio je " kulturnim i civiliziranim razgovorom".*

*" Došlo je do kratkih spojeva, jer neki očito nisu shvatili da je vrijeme predizborne kampanje prošlo", izjavio je Mesić.*

*Premijer je istaknuo.*

*" Ne vidim ozbiljne probleme i sukobe između predsjednika, Vlade, Sabora ili mene".*

*Na pitanje o promjeni Ustava, predsjedničkih ovlasti, itd., Račan je odgovorio da je riječ o većinskom opredjeljenju građana Hrvatske za parlamentarnu demokraciju i za izlazak iz predsjedničkog, odnosno polupredsjedničkog sustava.*

*On je istakao da nikako nije za ravnopravnu podjelu vlasti između Predsjednika, parlamenta i Vlade, jer bi" to doista govorilo o polupredsjedničkom ili poluparlamentarnom sustavu".*

*Za ulaganja veća od 20 milijuna kuna - upola manji porez na dobit.*

*Vlada priprema zakon kojim bi se za ulagače smanjili porezi ali i osigurala pomoć države.*

*Pred Vladom vrlo će se brzo naći nacrt zakona o poticanju ulaganja koji je pripremilo Ministarstvo gospodarstva koje je spremno izići pred Vladu s cijelim paketom pogodnosti - od poreznih olakšica i sudjelovanja u troškovima zapošljavanja, do korištenja državnih nekretnina po nižim cijenama ili čak bez naknade.*

*Na poticajne mjere ne bi, međutim, mogli računati nikakvi mutni poslovi, nego ulaganja veća od šest milijuna kuna, koja donose novu opremu i tehnologiju, uvode nove proizvode, otvaraju radna mjesta, povećavaju izvoz, šire ponudu usluga.*

*Zakon bi nudio povlastice i ako je investicija usmjerena na ona područja Hrvatske koja su sada po gospodarskoj snazi ispod državnog prosjeka.*

*Zakon bi se jednako odnosio na domaće i strane ulagače.*

*Ako bi bio prihvaćen prijedlog Ministarstva gospodarstva, država bi s privatnim ulagačima sufinancirala otvaranje novih radnih mjesta iznosom do 15 tisuća kuna godišnje po zaposlenom.*

*Za investicije veće od 20 milijuna kuna porez na dobit bio bi upola manji od uobičajene stope tijekom pet godina od početka ulaganja, uz uvjet da se zaposli najmanje 30 radnika.*

*Investitori koji ulože više od 40 milijuna kuna, a zaposle barem 50 radnika tijekom pet godina plaćali bi 34 posto propisane stope poreza na dobit.*

*Za ulaganja veća od 100 milijuna kuna porez na dobit smanjio bi se 80 posto, dok bi investicije iznad 400 milijuna kuna bile osam godina oslobođene poreza na dobit ako zaposle najmanje 200 radnika.*

*Odluke o poticajnim mjerama i poreznim povlasticama bile bi u nadležnosti povjerenstva za poticanje ulaganja.*

*Takvo povjerenstvo tek treba osnovati, a u njemu bi bio ministar gospodarstva, financija, rada i resora nadležnog za djelatnost u koju se ulaže.*

*S višečlanim povjerenstvom izbjegle bi se sumnje koje su dosad, i kada za to nije bilo razloga, padale na pojedinačne posrednike stranih investitora iz hrvatskih državnih struktura. (C W).*

*Kosovac nije smijenjen, Galić vraća mandat.*

*Glavni urednik HTV-a Obrad Kosovac nije smijenjen na sjednici Vijeća HRT-a, jer prijedlog generalnog direktora Mirka Galića o Kosovčevom razrješenju nije dobio natpolovičnu većinu glasova prisutnih članova Vijeća.*

*Galić je nakon toga izjavio da će vratiti svoj mandat na raspolaganje Saboru.*

*Direktor HRT-a ocijenio je da ga je tim potezom Vijeće na prvom koraku dezavuiralo, jer je ponudio program promjena za koje su potrebni novi ljudi, a dio članova Vijeća podržao je Kosovca koji se tome otvoreno suprotstavlja.*

*Članovi su potom zatražili od Galića da povuče svoju odluku o vraćanju mandata Saboru, jer je Vijeće jednoglasno podržalo njegov prijedlog programa.*

*Oni koji nisu poduprli Kosovčevu smjenu, obrazložili su to protivljenjem proceduralnim pogreškama u pripremi sjednice.*

*Prema Galićevu prijedlogu, nakon smjene Kosovca, bio bi imenovan vršitelj dužnosti glavnog urednika HTV-a do prihvatanja novog Zakona o HRT-u i raspisivanja javnog natječaja.*

*No, Kosovac je pred članovima Vijeća odbio podnijeti ostavku ili staviti mandat na raspolaganje, istaknuvši da ne osjeća krivnju niti sram za ono što je radio proteklih deset mjeseci, koliko je bio na dužnosti glavnog urednika HTV-a.*

*Nakon glasovanja (sedmero za Kosovčevu smjenu, troje protiv i petoro suzdržanih), novi predsjednik Vijeća Škrabalo kazao je da se takvom odlukom jako otežao položaj Galiću.*

*Predsjednik Vijeća HRT-a Ivo Škrabalo, izabran na sjednici u petak, najavio je da će 10 travnja sazvati novu sjednicu, na kojoj će jedina točka dnevnog reda biti prijedlog razrješenja glavnog urednika HTV-a.*

*Ivo Škrabalo je kasnije kazao kako je Vijeće HRT-a jednoglasno dalo potporu programu direktora Galića jer vodi prema javnoj televiziji.*

*Škrabalo je također kazao kako Galićevu nakanu vraćanja mandata Saboru nije shvatio kao najavu ostavke te da očekuje smjenu Obrada Kosovca 10 travnja.*

*To je politička smjena, kazao je Škrabalo.*

*Obrad Kosovac je profesionalac koji je svoje profesionalne sposobnosti stavio u neprimjerenu službu jedne politike i zbog toga učinio HTV takvom da je predmet svih natezanja između Hrvatske i međunarodne zajednice, naglašava Škrabalo.*

*Dobili smo novog direktora koji nije politička ličnost i koji u programu ima HTV osloboditi utjecaja politike, zaključio je novi predsjednik Vijeća Hrvatske radio televizije.*

*Daimler-Chrysler povećava ulaganja u Hrvatsku.*

*Već ove godine tvrtka Euroline-Mercedes investirat će u našu zemlju oko 20 milijuna njemačkih maraka, što će za početak omogućiti otvaranje 90 radnih mjesta.*

*Predstavnici Daimler-Chryslera, jednog od najvećih svjetskih koncerna, objavili su nakon razgovora s premijerom Ivicom Račanom i brojnim ministrima da žele povećati ulaganja u Hrvatsku.*

*Već ove godine tvrtka Euroline-Mercedes investirat će u našu zemlju oko 20 milijuna njemačkih maraka, što će za početak omogućiti otvaranje 90 radnih mjesta, izjavio je Mathias Kleiner, generalni opunomoćenik Daimler-Chryslera, nakon jednoipolsatnog sastanka u Vladi.*

*Predstavnici poslovnih jedinica Daimler-Chryslera još će detaljno razgovarati s predstavnicima Vlade i ministarstava kako bi se osmislili i započeli projekti koji bi trebali biti ostvareni prema prioritetima predloženim od hrvatske Vlade, najavio je Kleiner.*

*Koliko je taj sastanak bio značajan, govori podatak da je Daimler-Chrysler poslao u Zagreb delegaciju od čak 17 članova, koji predstavljaju isto toliko poslovnih jedinica tog koncerna.*

*Time je, po Kleinerovim riječima, koncern pokazao želju za jačanjem suradnje s Hrvatskom, kao i da želi pružiti podršku novoj hrvatskoj Vladi u ostvarivanju njezina programa, za što će joj, po njegovu mišljenju, biti potrebno puno odlučnosti i snage.*

*Inače, ta tvrtka sudjelovala je i na ovogodišnjem AutoShowu koji je prema rezultatima dosad najuspješnija priredba takve vrste u Hrvatskoj.*



## Prilog 2 – Dio zlatnog standarda za testiranje parsera

*Pakt konačno prohodao.*

*Ne vidim ozbiljne probleme i sukobe između predsjednika, Vlade, Sabora ili mene.*

*Daimler-Chrysler povećava ulaganja u Hrvatsku.*

*Zagrepčani piju zdravu prirodnu vodu.*

*Europska komisija financira čišćenje Dunava.*

*Slovenija će ekshumirati žrtve masovnih likvidacija iz porača.*

*Ta velika voda nekad je pokretala stotine mlinskih kotača.*

*Pritom ne bismo zanemarili kompetenciju drugih stručnjaka.*

*Hrvatske stolnotenisačice osvojile su europsku broncu.*

*Konferencija je polučila uspjeh iznad očekivanja - 2,47 milijardi eura za prvi paket obnove.*

*Premijer je istaknuo.*

*Zakon bi se jednako odnosio na domaće i strane ulagače.*

*Ali, želimo postaviti kriterije za ulazak u stranku.*

*Đukić traži autonomiju za Srbe u Hrvatskoj.*

*Čak i Mesić može biti odgovoran pred Haaškim sudom.*

*Prema tome, po načelu objektivne odgovornosti, čak i on može biti odgovoran pred Haaškim sudom.*

*Predsjednici svih vrsta, uozbiljite se!*

*Nacija može odahnuti.*

*Jer, pomireni su odmah nakon ručka pokazali novu vrstu neozbiljnosti.*

*U Bavarskoj su socijalni kršćani stvorili supermodernu ekonomiju uz očuvanje tradicionalnih društvenih vrednota.*

*Među predsjedničkim kandidatima s takvim programom bio je i sadašnji predsjednik Mesić.*

*On se ne zauzima za prekid ugovora s Enronom pod svaku cijenu.*

*Ni ministar gospodarstva Goranko Fižulić nije se sastao s predstavnicima EBRD-a.*

*Prva je nastavak izvršenja potpisanog ugovora.*

*Prodaja je koncem veljače u odnosu na isto lanjsko razdoblje povećana četiri posto.*

*Dvije velike bankarske krize stajale su Hrvatsku 5,5 milijardi dolara.*

*Bizovačke toplice su oaza slavonskog turizma.*

*Cijene su izuzetno povoljne.*

*Njezina temperatura od oko 12 stupnjeva idealna je za piće.*

*Zagreb leži na vodi.*

*Ova divizija nalazi se pod francuskim zapovjedništvom.*

*Šef portugalske diplomacije Jaime Gamma boravio je u Zagrebu.*

*Moramo staviti Hrvatsku na jasnu platformu za pregovore s Europskom unijom.*

*Vi polako ispunjavate uvjete za pristupanje.*

*Hrvatska mora čim prije podnijeti zahtjev za članstvo u EU.*

*Svoju treću knjigu na njemačkom jeziku Jergović će tako predstaviti u Koelnu, Berlinu, Hannoveru i Hamburgu te na lajpciškom sajmu knjiga.*

*Moćnik sv. Stjepana ipak ne odlazi u Mađarsku!*

*Upravo zbog te rijetkosti i stilske unikatnosti taj moćnik ne smije biti otuđen iz korpusa hrvatske kulturne baštine.*

*Ivana Jakupčević plasirala se u obaveznim sastavima ovogodišnjeg prvenstva svijeta u umjetničkom klizanju u završni nastup u slobodnim sastavima u Nici.*

*Šibenik, naime, u svojoj povijesnoj jezgri ima i najstarije sačuvane orgulje u Europi, te još tri dragocjena instrumenta u blizoj okolici.*

*Haaška tužiteljica u Zagrebu se susrela s predsjednikom Mesićem te premijerom Račanom.*

*Hrvatski predsjednik boravio je u dvodnevnom posjetu Makedoniji.*

*Tu vijest Tomčić je ocijenio kao jedan od prvih koraka do ulaska u NATO i EU.*

*Kako je prodan najtiražniji hrvatski dnevni list?*

*To je zločin.*

*To je izvrsno za Ameriku.*

*Đukić i Pupovac ne odustaju od stare retorike.*

*LS-ova politika" novog smjera" nije baš dobro prošla na parlamentarnim izborima.*



Posljednji parlamentarni izbori nisu bili izbori za stranke i njihove programe. Ta stabilizacija opet je strateški interes zapadnih vlada. Time je Granić svoje potencijalne poklonike stavio u dvojbu o ciljevima nove stranke i njezinoj političkoj različitosti u odnosu na ostatak ponude na političkoj sceni. Danas se ubraja u skupinu poznatijih ribičkih voda u ovom dijelu Europe. Jezero je bogato ribom (šarani, cipli, jegulje i druge ribe). Izložba će biti otvorena do konca svibnja. To je program za svaku pohvalu. Isključivo sam radila prema fotografijama iz prospekata. Razgovarali smo o dvjema konkretnim temama. Vesna Pusić predstavila je novi politički program. Zato su ponajprije politički odnosi dviju zemalja postali primjerom međusobne suradnje. Italija i Hrvatska pripremaju sporazum o strateškoj suradnji. Hrvatski premijer Ivica Račan razgovarao je 11 travnja s tajlandskom princezom o unapređenju gospodarskih odnosa dviju zemalja. Na brojnim pozornicama Torina ovih dana predstavljaju se kazališne skupine iz Grčke, Njemačke, Italije, Latvije, Rusije, Francuske i Finske. Unatoč tome, to djelo ostaje razumljivo u svojoj osnovnoj poruci. Tim izložbenim projektom Dimitrije Popović zaista pompozno predstavlja u Rimu suvremeno sakralno slikarstvo Hrvatske. Pri kraju procesije Juda simbolički pred svima okupljenima baca izdajnički novac. Od očekivanih turista hrvatski će turizam ove godine zaraditi 3,5 milijarde američkih dolara. Dobre vijesti stižu i iz Slovačke, Nizozemske, Austrije i Francuske. Pokazatelji s njemačkog tržišta još su bolji. Riječ je o vraćanju državnog duga gospodarstvu i građanima u visini od gotovo 10 milijardi kuna. Ovakvi potezi to potvrđuju. Hrvatska već neko vrijeme želi članstvo u Partnerstvu za mir. To treba biti razvijano u skladu s NATO-ovim standardima. Osnovana je Koordinacija za borbu protiv gospodarskog kriminala. Koordinacija će svoju prvu sjednicu imati već večeras. Pritužbe su stigle iz 310 hrvatskih mjesta, ponajviše iz Zagreba i Splita. Njegovu ostavku sada treba prihvatiti predsjednik Republike Stjepan Mesić. To želimo postići javnim natječajem. Treba čuvati dostojanstvo svake osobe. To je izraz bolesnog rješavanja problema. Međutim, nova vlast kao da kreće u suprotnom smjeru od bivše linijom ekstremizma. Pritom su se više puta sa saborske govornice čuli zahtjevi za Škrebom ostavkom. Prvi kilometri Krke nisu u granicama Nacionalnog parka Krka. Njegove dvije glavne stube imaju ukupnu visinu od 20 metara. Danas pokreće tek nekoliko njih, radi posjetitelja. Zaštita prirode uzrokovala je sukob šumara i Države oko zaštićenih područja. S takvim nerazumnim postupcima Hrvatske šume i šumarstvo nisu se nikad susreli u svojoj više od dva stoljeća dugoj povijesti. Tu" horor zaštitu" doista treba zaustaviti. U Hrvatskoj je proglašeno strašno puno zaštićenih područja. I tamo bi došlo do anarhičnog i kaotičnog ponašanja. Još uvijek smo traumatizirano društvo. Zajednica ima trenutačno oko stotinjak članova. Živimo uglavnom u Beirutu. U tim zemljama živi oko 400 tisuća hrvatskih iseljenika. Promatram svijet kukaca kao neku vrst svijeta snova, svijeta iz bajki. Uskoro će u dubrovačkom kraju list murve biti poput mišjega uha. Riječ je o tehnici slaganja krpica u mozaik. S tog hrvatskog prostora tijekom rata bilo je protjerano 20 tisuća Hrvata. Slične manifestacije održale su se i u drugim većim hrvatskim gradovima. Nadbiskup Bozanić naglasio je važnost pravedne naknade za rad za svakoga čovjeka, s posebnim naglaskom na mirovine. On bi morao biti ne samo izvan političkih stranaka nego i iznad njih. Jer Mesić bi se trebao baviti dubokom krizom hrvatskog društva. Francuski predsjednik dijeli vanjsku i sigurnosnu politiku sa svojim premijerom. Na osnivačkoj skupštini DC-a Granić izabran za predsjednika, a Vesna Škare-Ožbolt za zamjenicu. Od infrastrukturnih projekata najvažnija je financijska potpora dovršetku autoceste Zagreb - Goričan ( mađarska granica ).

### Prilog 3 – Opis imenskih paradigmi (hrCroNoun.nof)

#ugniježđeni grafovi – JEDNINA:

```

n021s = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + e/Voc+s + u/L+s + om/I+s;
n060s = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>če/Voc+s + u/L+s + om/I+s;
n041s = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + u/Voc+s + u/L+s + em/I+s;
n025s = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + u/Voc+s + u/L+s + om/I+s;
n0812s = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + u/Voc+s + <B1>že/Voc+s + u/L+s + om/I+s;
n022s = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s;
n001s = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + om/I+s;
n07s = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + <B1>že/Voc+s + u/L+s + om/I+s;
n09s = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + <B1>še/Voc+s + u/L+s + om/I+s;
n031s = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + u/Voc+s + u/L+s + em/I+s;
n0228s = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + u/Voc+s + u/L+s + om/I+s;
n65s = <E>/Nom+s + <B1>e/G+s + <B2>ci/D+s + <B1>(u/Acc+s + o/Voc+s) + <B2>ci/L+s + <B1>(om/I+s);
n50s = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + o/Voc+s + i/L+s + om/I+s);
n5061s = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + e/Voc+s + i/L+s + om/I+s);
n96s = <E>/Nom+s + <B1>(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + <B1>(u/L+s + om/I+s);
n968s = <E>/Nom+s + <B1>(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + <B1>(u/L+s + em/I+s);
n108s = <E>/Nom+s + <B1>(a/G+s + u/D+s + e/Acc+s + e/Voc+s + u/L+s + em/I+s);
n961s = <E>/Nom+s + <B1>(a/G+s + u/D+s + o/Acc+s + o/Voc+s + u/L+s + om/I+s);
n614s = <E>/Nom+s + <B1>(e/G+s + i/D+s) + <B2>ci/D+s + <B1>(u/Acc+s + o/Voc+s) + <B2>ci/L+s + <B1>(i/L+s + om/I+s);
n35s = <E>/Nom+s + <B2>r(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s);
n3521s = <E>/Nom+s + <B2>r(a/G+s + u/D+s) + <E>/Acc+s + <B2>r(e/Voc+s + u/L+s + om/I+s);
n391s = <E>/Nom+s + <B2>c(a/G+s + u/D+s) + <E>/Acc+s + <B2>c(u/Voc+s + u/L+s + em/I+s);

```

```

n77s = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + i/I+s + du/I+s;

#ugniježđeni grafovi - MNOŽINA

n021p = i/Nom+p + a/G+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p;

n074p = i/Nom+p + i/G+p + ima/D+p + i/Acc+p + i/Voc+p + ima/L+p + ima/I+p;

n060p = i/Nom+p + i/G+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p;

n65p = <B1>(e/Nom+p + a/G+p + ama/D+p + e/Acc+p + e/Voc+p + ama/L+p + ama/I+p);

n604p = <B1>(e/Nom+p + a/G+p + i/G+p) + <B2>aka/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

n614p = <B1>(e/Nom+p + a/G+p + i/G+p) + <B2>aka/G+p + ma/D+p + <B1>(e/Acc+p + i/Voc+p) + ma/L+p + ma/I+p;

n506p = <B1>e/Nom+p + <E>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

96p = a/Nom+p + a/G+p + ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p;

n096p = <B1>(a/Nom+p + a/G+p + <B1>ala/G+p + ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p);

n1003p = <B1>si/Nom+p + a/G+p + <B1>sima/D+p + e/Acc+p + <B1>s(i/Voc+p + ima/L+p + ima/I+p);

n081p = <B1>zi/Nom+p + a/G+p + <B1>zima/D+p + e/Acc+p + <B1>z(i/Voc+p + ima/L+p + ima/I+p);

n111p = <B1>ci/Nom+p + a/G+p + <B1>cima/D+p + e/Acc+p + <B1>ci(<E>/Voc+p + ma/L+p + ma/I+p);

n361p = <B2>(ci/Nom+p) + a/G+p + <B2>c(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

n0352p = <B2>r(i/Nom+p) + a/G+p + <B2>r(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

n02p = ov(:n021p);

n041p = ev(:n021p);

n3521p = <B2>rov(:n021p);

n015p = <B1>žev(:n021p);

n96p = <B1>(:96p);

APROp = <B1>(e/Nom+p + a/G+p + ima/D+p + e/Acc+p + e/Voc+p + ima/L+p + ima/I+p);

MUDRAC = :n001s + :n021p;

ČOVJEK = :n001s + <BW>ljud(:n060p);

BOG = :n07s + :n02p;

KNEZ = :n07s + :n015p;

ANESTEZIOLOG = :n07s + :n081p;

```

```

DUH = :n09s + ov(:n021p);

ZLODUH = :n09s + :n1003p;

SIBIR =:n021s;

BAT =:n021s + :n02p;

ALAT = :n021s + :n021p;

ANSAMBL = :n021s + :n021p + <B1>ala/G+p;

CVIJET = :n021s + <B4>jetov(:n021p);

PLIJEN =:n021s + <B4>ljenov(:n021p);

SLIJED = :n021s + <B4>jedov(:n021p);

ŽDRIJEB = :n021s + <B4>ebov(:n021p) + <B4>jebov(:n021p);

SAT =:n021s + :n060p;

GOLUB = :n022s + :n02p;

ASTRONOM = :n022s + :n021p;

MED = :n025s;

GLAS = :n025s + :n02p;

PRIMJER = :n025s + :n021p;

LEŠ = :n025s + :n041p;

BRATIĆ = :n031s + :n021p;

KRALJ = :n031s + :n041p;

MINISTAR = :n35s + :n0352p;

TIGAR = :n35s + :n3521p;

BOŽIĆ = :n041s;

BESKRAJ = :n041s + :n021p;

BRIDŽ = :n041s + :n041p;

VIC = :n041s + om/I+s + :n041p;

BRAĆA = :n50s;

AGRONOMKA = :n50s + :n604p;

ABECEDA = :n50s + :n65p;

MAGLA = :n50s + :n65p + <B1>i/G+p + <L2>a/G+p;

NARANČA = :n50s + :n65p + <B1>i/G+p + <L2>a/G+p;

KROŠNJA = :n50s + :n65p + <B1>i/G+p + <B3>anja/G+p;

```

```

DUBROVNIK = :n060s;

VLAK = :n060s + :n02p;

MJESEC = :n060s + :n060p;

BAJKA = :n65s + :n604p;

AMERIKA = :n65s + :n65p;

GAMAD = :n77s;

GLAD      = :n77s + :n074p;

EURO      = :n96s + <B1>(:n02p);

BLAGO     = :n96s + :n96p;

STABLO    = :n96s + :n096p;

BILJE     = :n108s;

CRTANJE   = :n108s + :n96p;

NEPCE     = :n108s + :n96p + <B1>a<L2><B1>ba/G+p;

LEPTIR    = :n0228s + :n021p;

KOTAR     = :n0228s + :n041p;

JEZIČAC   = :n391s + :n361p;

ZBIRKA    = :n614s + :n614p;

BIJEG     = :n0812s + :n02p;

PRIJEDLOG = :n0812s + :n081p;

SJEMENJE  = :n968s + :n96p;

METAR     = :n3521s + :n0352p;

KADAR     = :n3521s + :n3521p;

PROLAZNIK = :n001s + :n111p;

AKT       = :n021s + i/p+Nom + a/p+G + a<L2>a/p+G + ima/p+D + e/p+Acc + i/p+Voc +
ima/p+L + ima/p+I;

DIJAMANT  = :n021s + i/Nom+p + a/G+p + <B1>ata/G+p + ima/D+p + e/Acc+p + i/Voc+p +
ima/L+p + ima/I+p;

AGENT     = :n022s + i/Nom+p + a/G+p + <B1>ata/G+p + ima/D+p + e/Acc+p + i/Voc+p +
ima/L+p + ima/I+p;

ARGATIN   = :n022s + <B2>(i/p+Nom + a/p+G + ima/p+D + e/p+Acc + i/p+Voc + ima/p+L
+ ima/p+I);

GOSPODIN  = :n022s + <B2>(a/Nom+p + e/G+p + i/D+p + u/Acc+p + o/Voc+p + i/L+p +
om/I+p);

PRST      = :n022s + i/Nom+p + a/G+p + i/G+p + iju/G+p + ima/D+p + e/Acc+p +
i/Voc+p + ima/L+p + ima/I+p;

```

AMBULANTA = :n50s + <B1>(e/Nom+p + i/G+p + <B1>ata/G+p) + <E>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

CIGLA = :n50s + <B1>(e/Nom+p + i/G+p) + <E>/G+p + <B2>ala/G+p + <B1>(ama/D+p + e/Acc+p + e/Voc+p + ama/L+p + ama/I+p);

CRKVA = :n50s + <B1>(e/Nom+p + i/G+p) + <E>/G+p + <B2>ava/G+p + <B1>(ama/D+p + e/Acc+p + e/Voc+p + ama/L+p + ama/I+p);

GRIŽNJA = :n50s + <B1>(e/Nom+p + i/G+p) + <E>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

IGRA = :n50s + <B1>(e/Nom+p + i/G+p) + <E>/G+p + <L2>a/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

LUTKA = :n50s + <B1>(e/Nom+p) + <B2>aka/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

NAREDBA = :n50s + <B1>(e/Nom+p + i/G+p) + a/G+p + <B2>aba/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

PJESMA = :n50s + <B1>e/Nom+p + <L2>a/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

USNA = :n50s + <B1>(e/Nom+p + i/G+p + <B1>ana/G+p) + <E>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

ZEMLJA = :n50s + <B1>(e/Nom+p + i/G+p + <B2>alja/G+p) + <E>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

OVCA = :n50s + <B1>e/Nom+p + <L2>a/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p + ma/L+p + ma/I+p);

OTOK = :n060s + :n111p;

PISMO = :n96s + <B1>(a/Nom+p + a/L2>a/G+p + ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p);

OKO = :n96s + <B2>či(<E>/Nom+p + ju/G+p + ma/D+p + <E>/Acc+p + <E>/Voc+p + ma/L+p + ma/I+p);

UHO = :n96s + <B2>ši(<E>/Nom+p + ju/G+p + ma/D+p + <E>/Acc+p + <E>/Voc+p + ma/L+p + ma/I+p);

POPIJEVKA = :n614s + <B1>(e/Nom+p + i/G+p) + <B2>aka/G+p + <B2>aka<LW><R3><S1>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

TRAG = :n0812s + ov(i/Nom+p + a/G+p + ima/D+p + e/Acc+p + e/Voc+p + ima/L+p + ima/I+p);

ČOVJEČANSTVO = :n961s + <B1>(a/Nom+p + a/G+p) + <B2>ava/G+p + <B1>(ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p);

ZLO = :n961s + <B1>a/Nom+p + <B2>ala/G+p + <B1>(ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p);

DRVCE = :n968s + <B1>(a/Nom+p + a/G+p) + <B2>aca/G+p + <B1>(ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p);

SRCE = :n968s + <B1>(a/Nom+p + a/G+p + <B1>daca/G+p + ima/D+p + a/Acc+p + a/Voc+p + ima/L+p + ima/I+p);

DISK = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B2>šće/Voc+s + u/Voc+s + u/L+s + om/I+s + :n02p;

LIK = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + u/Voc+s + če/Voc+s + u/L+s + om/I+s + :n02p;

VRH = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>še/Voc+s + u/Voc+s + u/L+s + om/I+s + :n02p;

OVAN = <E>/Nom+s + <B2>n(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s + ov(:n021p));

SAN = <E>/Nom+s + <B2>n(a/G+s + u/D+s) + <E>/Acc+s + <B2>n(u/Voc+s + u/L+s + om/I+s + ov(:n021p));

ŽIRI = <E>/Nom+s + ja/G+s + ju/D+s + <E>/Acc+s + <E>/Voc+s + ju/L+s + jem/I+s + j(:n021p);

ATAŠE = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + u/Voc+s + <E>/Voc+s + u/L+s + om/I+s + :n021p;

ATELJE = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <E>/Voc+s + u/L+s + om/I+s + :n021p;

BUNAR = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + u/Voc+s + e/Voc+s + u/L+s + em/I+s + om/I+s + :n021p;

KOŠTAC = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s + :n021p;

TROŠAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>če/Voc+s + <B2>k(u/Voc+s + u/L+s + om/I+s + ov(:n021p));

VOSAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>šče/Voc+s + <B2>k(u/Voc+s + u/L+s + om/I+s) + <B2>kov(:n021p);

OTAC = <E>/Nom+s + <B3>c(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s) + <B3>čev(:n021p);

PIJETAO = <E>/Nom+s + <B2>l(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s) + <B2>lov(:n021p);

PREDIO = <E>/Nom+s + <B2>jel(a/G+s + u/D+s) + <E>/Acc+s + <B2>jel(e/Voc+s + u/L+s + om/I+s) + <B2>jel(:n021p);

PRINC = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s + <B1>čev(:n021p);

BIORITAM = <E>/Nom+s + <B2>m(a/G+s + u/D+s) + <E>/Acc+s + <B2>m(e/Voc+s + u/L+s + om/I+s + ov(:n021p));

DAVALAC = <E>/Nom+s + <B3>oc(a/G+s + u/D+s + a/Acc+s) + <B3>oče/Voc+s + <B3>oc(u/L+s + em/I+s + <B3>oc(:n021p);

DIO = <E>/Nom+s + <B1>jel(a/G+s + u/D+s) + <E>/Acc+s + <B1>jel(e/Voc+s + u/L+s + om/I+s + ov(:n021p));

GLEŽANJ = <E>/Nom+s + <B3>nj(a/G+s + u/D+s) + <E>/Acc+s + <B3>nj(u/Voc+s + u/L+s + em/I+s + ev(:n021p));

CAR = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + em/I+s + :n041p;

SVRHA = <E>/Nom+s + <B1>e/G+s + <B2>si/D+s + <B1>(u/Acc+s + o/Voc+s) + <B2>si/L+s + <B1>(om/I+s) + :n65p;

```

BIT      = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + i/I+s +
ću/I+s      + :n074p;

BLAGOST  = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + <B2>šču/I+s +
i/I+s      + :n074p;

BOL      = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + ju/I+s +
i/I+s      + :n074p;

LJUBAV  = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + lju/I+s +
i/I+s      + :n074p;

MOĆ      = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + u/I+s      +
:n074p;

KOST     = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + <B2>šču/I+s +
:n074p + iju/G+p;

JAJE     = <E>/Nom+s + ta/G+s + tu/D+s + <E>/Acc+s + <E>/Voc+s + tu/L+s + tom/I+s
+          :n96p;

KONOPAC  = <E>/Nom+s + <B2>c(a/G+s + u/D+s) + <E>/Acc+s + <B2>(če/Voc+s + cu/L+s +
cem/I+s)    + :n361p;

TRGOVAC  = <E>/Nom+s + <B2>(c(a/G+s + u/D+s + a/Acc+s) + če/Voc+s + cu/L+s +
cem/I+s)    + :n361p;

PISAC    = <E>/Nom+s + <B2>c(a/G+s + u/D+s + a/Acc+s) + <B3>šče/Voc+s +
<B2>c(u/L+s + em/I+s)      + :n361p;

SVETAC   = <E>/Nom+s + <B2>c(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s +
em/I+s)    + <B3>c(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s)

+ :n361p

+ <B3>c(i/Nom+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

BRNJICA  = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + e/Voc+s + o/Voc+s + i/L+s +
om/I+s)    + :n506p;

DUGA     = <E>/Nom+s + <B1>(e/G+s + i/D+s) + <B2>zi/D+s + <B1>(u/Acc+s + o/Voc+s +
i/L+s) + <B2>zi/L+s + <B1>(om/I+s      + :n506p;

GUSJENICA = :n5061s      + :n506p;

KNJIGA   = <E>/Nom+s + <B1>(e/G+s + <B1>zi/D+s + u/Acc+s + o/Voc+s + <B1>zi/L+s +
om/I+s)    + :n506p;

JEKA     = <E>/Nom+s + <B1>(e/G+s + i/D+s) + <B2>ci/D+s + <B1>(u/Acc+s + o/Voc+s +
i/L+s + om/I+s) + <B2>ci/L+s      + :n506p;

ZALIHA   = <E>/Nom+s + <B1>(e/G+s + i/D+s + si/D+s + u/Acc+s + o/Voc+s + i/L+s +
si/L+s + om/I+s)      + :n506p;

NEUSPJEH = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + u/Voc+s + <B1>še/Voc+s + u/L+s
+ om/I+s      + :n1003p;

PROPUH   = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>še/Voc+s + u/L+s + om/I+s
+ :n1003p;

ADVENTIZAM = <E>/Nom+s + <B2>m(a/G+s + u/D+s) + <E>/Acc+s + <B2>m(e/Voc+s + u/L+s +
om/I+s +

i/Nom+p) + a/G+p + <B2>m(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

```



AFRIKA = <E>/Nom+s + <B1>e/G+s + <B2>ci/D+s + <B1>u/Acc+s + <B1>o/Voc+s + <B2>ci/L+s + <B1>om/I+s;

AUSTRALIJA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + a/Voc+s + o/Voc+s + i/L+s + om/I+s);

ANĐEO = <E>/Nom+s + <B1>l(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s + i/p+Nom + a/p+G + ima/p+D + e/p+Acc + i/p+Voc + ima/p+L + ima/p+I);

BOJAZAN = <E>/Nom+s + <B2>n(i/G+s + i/D+s) + <E>/Acc+s + <B2>n(i/Voc+s + i/L+s + i/I+s) + <B3>žnju/I+s + <B2>n(:n074p);

BRLOG = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>že/Voc+s + u/L+s + om/I+s;

CIPAL = <E>/Nom+s + <B2>l(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s + i/Nom+p) + a/G+p + <B2>l(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

DIJETE = <E>/Nom+s + ta<L7><S>/G+s + tu<L7><S>/D+s + <E>/Acc+s + <E>/Voc+s + tu<L7><S>/L+s + tom<L8><S>/I+s + <B5>jec(a/Nom+p + e/G+p + i/D+p + u/Acc+p + o/Voc+p + i/L+p + om/I+p);

DOBA = <E>/Nom+s + <E>/G+s + <B1>(u/D+s + om/Acc+s + <E>/Voc+s + <B1>(u/L+s + om/I+s) + <E>/Nom+p + <E>/G+p + <B1>ima/D+p + <E>/Acc+p + <E>/Voc+p + <B1>(ima/L+p + ima/I+p);

DOHODAK = <E>/Nom+s + <B3>tk(a/G+s + u/D+s) + <E>/Acc+s + <B2>če/Voc+s + <B3>če/Voc+s + <B3>tk(u/Voc+s + u/L+s + om/I+s) + <B3>ci/Nom+p + <B2>cu/Nom+p + a/G+p + <B3>cim/D+p + <B2>cima/D+p + <B3>tke/Acc+p + <B2>ci/Voc+p + <B3>ci/Voc+p + <B3>(cima/L+p + cima/I+p) + <B2>(cima/L+p + cima/I+p);

DORUČAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>k(u/Voc+s + u/L+s + om/I+s + ci/Nom+p) + a/G+p + <B2>(cima/D+p + ke/Acc+p + ci/Voc+p + cima/L+p + cima/I+p);

DVOJICA = <E>/Nom+p + <B1>(e/G+p + i/D+p + u/Acc+p + o/Voc+p + i/L+p + om/I+p) + <E>/Voc;

FINANCIJE = <E>/Nom+p + <B1>(a/G+p + ama/D+p) + <E>/Acc+p + <E>/Voc+p + <B1>(ama/L+p + ama/I+p);

FRANCUSKA = <E>/Nom+s + <B1>(e/G+s + oj/D+s + u/Acc+s) + <E>/Voc+s + <B1>o(j/L+s + m/I+s);

GREŠKA = <E>/Nom+s + <B1>(e/G+s + i/D+s) + <B2>ci/D+s + <B1>(u/Acc+s + o/Voc+s + i/L+s) + <B2>ci/L+s + <B1>(om/I+s + e/Nom+p + a/G+p + i/G+p) + <L2>a/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

KOKOŠ = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + i/I+s + ju/I+s + i/Nom+p + i/G+p + iju/G+p + ima/D+p + i/Acc+p + i/Voc+p + ima/L+p + ima/I+p;

IME = <E>/Nom+s + na/G+s + nu/D+s + <E>/Acc+s + <E>/Voc+s + nu/L+s + nom/I+s

```

+ n(:96p);

JUTRO = <E>/Nom+s + <B1>(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + <B1>u/L+s + m/I+s
+ <B1>a/Nom+p + <B2>ara/G+p + <B1>(ima/D+p + a/Acc+p + a/Voc+p + ima/L+p +
ima/I+p);

MISAO = <E>/Nom+s + <B2>l(i/G+s + i/D+s) + <E>/Acc+s + <B2>l(i/Voc+s + i/L+s +
i/I+s + <B1>šlju/I+s) + <B2>l(:n074p);

NOGA = <E>/Nom+s + <B1>e/G+s + <B2>zi/D+s + <B1>(u/Acc+s + o/Voc+s) + <B2>zi/L+s +
<B1>(om/I+s
+ e/Nom+p + u/G+p) + <E>/G+p + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

PODNE = <E>/Nom+s + v(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + v(u/L+s + om/I+s)
+ v(:96p);

PODRŠKA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + o/Voc+s + i/L+s + om/I+s) +
<B2>(ci/G+s + ci/L+s)
+ <B1>(e/Nom+p + i/G+p + a/G+p + ama/D+p + e/Acc+p + e/Voc+p + ama/L+p +
ama/I+p) + <B3>žaka/G+p;

POSAO = <E>/Nom+s + <B2>l(a/G+s + u/D+s) + <E>/Acc+s + <B2>l(e/Voc+s + u/L+s +
om/I+s
+ ovi/Nom+p + ova/G+p + ovima/D+p + ove/Acc+p + ovi/Voc+p + ovima/L+p +
ovima/I+p);

OSTATAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>če/Voc+s +
<B3>če/Voc+s + <B2>k(u/Voc+s + u/L+s + om/I+s)
+ <B3>ci/Nom+p + <B2>ci/Nom+p + a/G+p + <B3>cima/D+p + <B2>cima/D+p +
<B2>ke/Acc+p + <B2>ci/Voc+p + <B3>ci/Voc+p + <B3>cima/L+p + <B2>cima/L+p +
<B3>cima/I+p + <B2>cima/I+p;

PIJESAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>ku/Voc+s +
<B3>šče/Voc+s + <B2>k(u/L+s + om/I+s)
+ <B2>ci/Nom+p + a/G+p + <B2>cima/D+p + <B2>(ke/Acc+p + ci/Voc+p + cima/L+p
+ cima/I+p);

RADIO = <E>/Nom+s + <B1>j(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + <B1>j(u/L+s +
em/I+s) + <B1>j(:n021p);

TJEDAN = <E>/Nom+s + <B2>n(a/G+s + u/D+s) + <E>/Acc+s + <B2>n(e/Voc+s + u/L+s +
om/I+s
+ i/Nom+p) + a/G+p + <B2>n(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

UZORAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>če/Voc+s + <B2>k(u/L+s
+ om/I+s)
+ <B2>ci/Nom+p + a/G+p + <B2>cima/D+p + <B2>(ke/Acc+p + ci/Voc+p + cima/L+p
+ cima/I+p);

SUDAC = <E>/Nom+s + <B3>c(a/G+s + u/D+s + a/Acc+s) + <B3>če/Voc+s + <B3>c(u/L+s +
em/I+s
+ i/Nom+p) + a/G+p + <B3>c(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p)
+ <B2>c(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s

```

```

+ i/Nom+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

UOBRAŽENKO = :FABIANO

+ <B2>(aka/Nom+p + aka/G+p + cima/D+p + ke/Acc+p + c(i/Voc+p + ima/L+p + ima/I+p));

UOBRAŽENKA = <E>/Nom+s + <B1>e/G+s + <B2>ci/D+s + <B1>(u/Acc+s + a/Voc+s + i/L+s +
om/I+s)

+ <B1>(e/Nom+p + <B1>aka/G+p) + ma/D+p + <B1>(e/Acc+p + e/Voc+p) + ma/L+p + ma/I+p;

UŽE = <E>/Nom+s + t(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + t(u/L+s + om/I+s)

+ <B1>ad(<E>/Nom+p + i/G+p + i/D+p + <E>/Acc+p + i/Voc+p + i/L+p + i/I+p) +
<B1>ađu/I+p;

VRATA = <E>/Nom+p + <E>/G+p + <B1>ima/D+p + <E>/Acc+p + <E>/Voc+p + <B1>(ima/L+p +
ima/I+p);

VRIJEME = <E>/Nom+s + <B5>emen(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s +
<B5>emen(u/L+s + om/I+s) + <B5>emen(:96p);

ZALAZAK = <E>/Nom+s + <B3>sk(a/G+s + u/D+s) + <E>/Acc+s + <B3>(sku/Voc+s +
šče/Voc+s + sku/L+s + skom/I+s

+ sci/Nom+p) + a/G+p + <B3>s(cima/D+p + ke/Acc+p + ci/Voc+p + cima/L+p +
cima/I+p);

ZAGONETKA = <E>/Nom+s + <B1>(e/G+s + i/D+s + <B1>ci/D+s + <B2>ci/D+s + u/Acc+s +
o/Voc+s + i/L+s + <B1>ci/L+s + <B2>ci/L+s + om/I+s

+ e/Nom+p + i/G+p + <B1>aka/G+p) + <E>/G+p + ma/D+p + <B1>(e/Acc+p +
e/Voc+p) + ma/L+p + ma/I+p;

ŽABAC = <E>/Nom+s + <B3>pc(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s

+ i/Nom+p) + a/G+p + <B3>pc(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

ŽENSKA = <E>/Nom+s + <B1>(e/G+s + oj/D+s + u/Acc+s) + <E>/Voc+s + <B1>(oj/L+s +
om/I+s + e/Nom+p + ih/G+p + ima/D+p + e/Acc+p + e/Voc+p + ima/L+p + ima/I+p);

IZLOŽAK = <E>/Nom+s + <B3>šk(a/G+s + u/D+s) + <E>/Acc+s + <B3>š(če/Voc+s +
ku/Voc+s + ku/L+s + kom/I+s +

ci/Nom+p) + a/G+p + <B3>š(cima/D+p + ke/Acc+p + ci/Voc+p + cima/L+p + cima/I+p);

KĆI = <E>/Nom+s + <B1>er(i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + ju/I+s +
i/I+s) + <B1>er(:n074p);

LIPANJ = <E>/Nom+s + <B3>nj(a/G+s + u/D+s) + <E>/Acc+s + <B3>nj(u/Voc+s + u/L+s +
em/I+s +

i/Nom+p) + a/G+p + <B3>nj(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

MAESTRO = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s + o/Voc+s + u/L+s + om/I+s +

i/Nom+p + a/G+p) + <B2>ara/G+p + <B1>(ima/D+p + e/Acc+p + i/Voc+p +
ima/L+p + ima/I+p);

MOMAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s + a/Acc+s) + <B2>če/Voc+s + <B2>k(u/L+s +
om/I+s)

```

+ <B2>ci/Nom+p + a/G+p + <B2>cima/D+p + <B2>(ke/Acc+p + ci/Voc+p + cima/L+p + cima/I+p);

PAS = <E>/Nom+s + <B2>s(a/G+s + u/D+s + a/Acc+s + u/Voc+s + u/L+s + om/I+s + i/Nom+p) + a/G+p + <B2>s(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

STOŽAC = <E>/Nom+s + <B3>šc(a/G+s + u/D+s) + <E>/Acc+s + <B3>š(če/Voc+s + cu/L+s + cem/I+s) + <B3>šci/Nom+p + a/G+p + <B3>šc(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

TISAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B3>šče/Voc+s + <B2>k(u/L+s + om/I+s);

ADELAIDE = <E>/Nom+s + <E>/G+s + <E>/D+s + <E>/Acc+s + <E>/Voc+s + <E>/L+s + <E>/I+s;

AGE = <E>/Nom+s + <E>/G+s + <B1>(i/D+s + u/Acc+s) + <E>/Voc+s + <B1>(i/L+s + om/I+s);

AASEF = :n022s;

ABDERIĆANIN = :n022s + <B1>/Nom+p + <B2>a/G+p + <B1>ma/D+p + <B2>e/Acc+p + <B1>(<E>/Voc+p + ma/L+p + ma/I+p);

ADILJ = :n031s;

ABID = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + e/Voc+s + <E>/Voc+s + u/L+s + om/I+s;

ŽELIDRAG = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + <B1>že/Voc+s + u/L+s + om/I+s;

AFGANISTAN = :n021s;

ANGELES = :n025s;

BIHAĆ = :n041s;

BRIBIR = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + e/Voc+s + u/Voc+s + u/L+s + om/I+s;

ALLEN = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s;

ATLANTIK = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>če/Voc+s + u/L+s + om/I+s;

ANNECY = <E>/Nom+s + j(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + j(u/L+s + em/I+s);

ŽELJEZNO = <E>/Nom+s + g/G+s + ga/G+s + m/D+s + me/D+s + mu/D+s + <E>/Acc+s + <E>/Voc+s + m/L+s + me/L+s + mu/L+s + <B1>im/I+s;

ABELE = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + <B1>(u/L+s + om/I+s);

ADAMO = :ABELE + <B1>(:n021p);

ANđEO = <E>/Nom+s + <B1>l(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + <B1>l(u/L+s + om/I+s);

ABERTO = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + <B1>u/L+s + m/I+s;

ACIJA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + o/Voc+s) + <E>/Voc+s + <B1>(i/L+s + om/I+s);

ABA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s) + <E>/Voc+s + <B1>(i/L+s + om/I+s);

ABAZA = :ABA + :n506p;

ADICA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + e/Voc+s + i/L+s + om/I+s);

ALJASKA = <E>/Nom+s + <B1>(e/G+s + i/D+s) + <B2>ci/D+s + <B1>(u/Acc+s + o/Voc+s) + <B2>ci/L+s + <B1>(i/L+s + om/I+s);

AVELLINO = <E>/Nom+s + <B1>(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s + <B1>u/L+s + m/I+s;

BORNEO = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s) + <E>/Acc+s + <E>/Voc+s + <B1>(u/L+s + om/I+s);

ADELIO = <E>/Nom+s + <B1>(a/G+s + ja/G+s + ju/D+s + u/D+s + a/Acc+s + ja/Acc+s) + <E>/Voc+s + <B1>(ju/L+s + u/L+s + om/I+s + jem/I+s);

ADRIA = <E>/Nom+s + <B1>(e/G+s + je/G+s + i/D+s + ji/D+s + ju/Acc+s + u/Acc+s) + <E>/Voc+s + <B1>(i/L+s + ji/L+s + jom/I+s + om/I+s);

ABDULOVSKA = :FRANCUSKA

+ <B1>(e/Nom+p + ih/G+p + ima/D+p + e/Acc+p + e/Voc+p + ima/L+p + ima/I+p);

BRATISLAVA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s) + <E>/Voc+s + <B1>(o/Voc+s + i/L+s + om/I+s);

ABDIJA = :BRATISLAVA

+ <B1>e/Nom+p + <E>/G+p + <B1>(ima/D+p + e/Acc+p + e/Voc+p + ima/L+p + ima/I+p);

BRESCIA = <E>/Nom+s + <B1>e/G+s + <B1>/D+s + <B1>u/Acc+s + <E>/Voc+s + <B1>o/Voc+s + <B1>/L+s + <B1>om/I+s;

ŽELJENI = <E>/Nom+s + <B1>o(g/G+s + m/D+s + me/D+s + g/Acc+s) + <E>/Voc+s + <B1>o(me/L+s + m/L+s) + m/I+s + me/I+s;

ABDULOVSKI = :ŽELJENI

+ <E>/Nom+p + h/G+p + ma/D+p + <B1>e/Acc+p + <E>/Voc+p + ma/L+p + ma/I+p;

ALEKSANDAR = <E>/Nom+s + <B2>r(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s);

ANđELAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s + a/Acc+s) + <B2>če/Voc+s + <B2>k(u/L+s + om/I+s);

ŽUMBERAK = <E>/Nom+s + <B2>k(a/G+s + u/D+s) + <E>/Acc+s + <B2>če/Voc+s + <B2>k(u/L+s + om/I+s);

ĐURAD = <E>/Nom+s + <B2>đ(a/G+s + u/D+s + a/Acc+s + u/Voc+s + u/L+s + em/I+s);

ČAKOVEC = <E>/Nom+s + <B2>c(a/G+s + u/D+s) + <E>/Acc+s + <B2>če/Voc+s + <B2>c(u/L+s + em/I+s);

ABIANAC = <E>/Nom+s + <B2>c(a/G+s + u/D+s + a/Acc+s) + <B2>če/Voc+s + <B2>c(u/L+s + em/I+s)

```

+ <B2>ci/Nom+p + a/G+p + <B2>c(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

ŽDRIJEBE = <E>/Nom+s + <B5>ebet(a/G+s + u/D+s) + <E>/Acc+s + <E>/Voc+s +
<B5>ebet(u/L+s + om/I+s);

ABDILBAKI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + j(u/L+s +
om/I+s);

ABDURAIMI = :ABDILBAKI + j(:n021p);

ABAFFY = :ABDILBAKI + j(:n021p);

ŽORŽI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + j(u/L+s +
em/I+s);

ABDILFARUK = :n001s;

ADAMIK = :n001s + n111p;

ABDULAH = :n0228s;

ABDULAHU = :ARNE + :n021p;

ŽAKLIN = :ADELAIDE + :AASEF;

ADILJ = :n031s;

ACO = :n50s;

ADAMENKO = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s + o/Voc+s + u/L+s + >om/I+s) +
<B1>(:n021p);

AEROPISTA = :n50s;

AFRIKA = :n65s;

AFRIKANKA = :n50s

+ <B1>(e/Nom+p + a/G+p + i/G+p + ama/D+p + e/Acc+p + e/Voc+p + ama/L+p + ama/I+p);

ALAH = :n09s

+ <B1>(si/Nom+p + ha/G+p + sima/D+p + he/Acc+p + si/Voc+p + sima/L+p + sima/I+p);

ALAICA = :n5061s + :n65p;

ALAJBEG = :n07s + :n081p;

ALBANEZE = :n022s + <B1>(:n021p);

ALEKSENDER = :n35s;

ALHAJ = :n031s + :n021p;

ALMER = :n031s + om/I+s + :n021p;

AMBROŽ = :n022s + u/Voc+s + em/I+s + :n021p;

ANDY = <E>/Nom+s + <E>/G+s + ja/G+s + <E>/D+s + ju/D+s + <E>/Acc+s + ja/Acc+s +
<E>/Voc+s + <E>/L+s + ju/L+s + <E>/I+s + jom/I+s;

```

```

ANE = :n5061s;

ANTOLEC = <E>/Nom+s + <B2>(ca/G+s + cu/D+s + ca/Acc+s + ec/Voc+s + če/Voc+s +
cu/L+s + em/I+s) +

<B2>(ci/Nom+p + aca/G+p + cima/D+p + ce/Acc+p + ci/Voc+p + cima/L+p + cima/I+p);

APRO = :n50s + :APROp;

ARNE = <E>/Nom+s + a/G+s + u/D+s + a/Acc+s + <E>/Voc+s + u/L+s + om/I+s;

ARTEMIJE = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + em/I+s);

ASHDOD = :n021s + :n021p;

ATENA = :n50s + a/Voc+s + :n65p;

AUDI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + <E>/Acc+s + <E>/Voc+s + j(u/L+s +
om/I+s + em/I+s) + j(:n021p);

BABORSKY = <E>/Nom+s + <B1>o(ga/G+s + mu/D+s + ga/Acc+s) + <E>/Voc+s +
<B1>(ome/L+s + im/I+s) +

<B1>(i/Nom+p + ih/G+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p +
ima/I+p) + ji(<E>/Nom+p + <E>/Voc+p + ma/L+p + ma/I+p);

BAHRIJA = :n50s + a/Voc+s;

BAJDE = :n5061s + :n65p;

BAK = :n031s + <E>/Voc+s + om/I+s + :n021p;

BALATON = :n021s + a/Acc+s + :n021p;

BALAŽ = :n022s + em/I+s + :n021p;

BASK = :n0228s + :n021p;

BATTHYANI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + j(u/L+s + em/I+s)
+ j(:n021p);

BERAM = <E>/Nom+s + <B2>m(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s)
+ <B2>mi/Nom+p + a/G+p + <B2>m(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p +
ima/I+p);

BERI = <E>/Nom+s + <E>/G+s + <E>/D+s + <E>/Acc+s + <E>/L+s + <E>/I+s + j(a/G+s +
u/D+s + a/Acc+s) + <E>/Voc+s + j(u/L+s + om/I+s)
+ j(:n021p);

BLAGAJ = :n041s + a/Acc+s + :n021p;

BLAGOVIJEST = <E>/Nom+s + i/G+s + i/D+s + <E>/Acc+s + i/Voc+s + i/L+s + i/I+s +
<B2>šću/I+s + :n074p;

BLAIR = :n022s + u/Voc+s + :n021p;

BORICA = :n5061s;

BOSANKA = :n50s + a/Voc+s + :n65p + i/G+p;

```

BOŽAC = <E>/Nom+s + <B3>šc(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s)  
+ <B3>šci/Nom+p + a/G+p + <B3>šc(ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

BRAZZAVILLE = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <E>/Voc+s + u/Voc+s + u/L+s  
+ om/I+s;

BROZ = :n022s + e/Voc+s + :n021p;

BRČKO = <E>/Nom+s + g/G+s + ga/G+s + me/D+s + m/D+s + mu/D+s + <E>/Acc+s +  
<E>/Voc+s + m/L+s + me/L+s + mu/L+s + <B1>im/I+s;

BUJAČENKO = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s + o/Voc+s + u/L+s + m/I+s)  
+ <B2>(ci/Nom+p + ka/G+p + cima/D+p + ke/Acc+p + ci/Voc+p + cima/L+p + cima/I+p);

BUNJEVKA = :n50s  
+ <B2>(ke/Nom+p + aka/G+p + ki/G+p + kama/D+p + ke/Acc+p + ka/Voc+s +  
ke/Voc+p + ama/L+p + kama/I+p);

BUTIŠNICA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + o/Voc+s + e/Voc+s + i/L+s +  
om/I+s);

CARIN = :n022s + <E>/G+s + <E>/D+s + <E>/Acc+s + <E>/Voc+s + <E>/L+s + <E>/I+s +  
:n021p;

CIGANČINA = :n50s + a/Voc+s + :n65p;

CORNELIO = <E>/Nom+s + <B1>(a/G+s + ja/G+s + ju/D+s + u/D+s + a/Acc+s + ja/Acc+s +  
o/Voc+s + ju/L+s + u/L+s + jom/I+s + om/I+s);

CROATIA = <E>/Nom+s + <B1>(ja/Nom+s + je/G+s + ji/D+s + ju/Acc+s + a/Voc+s +  
ja/Voc+s + jo/Voc+s + ji/L+s + jom/I+s);

CVJETNICA = :n50s + :n65p;

DANČE = :n5061s + e/D+s + e/Acc+s + e/L+s + e/I+s;

DAVOR = :n022s;

DELIPETAR = :n35s + :n0352p;

DESIRE = <E>/Nom+s + <E>/G+s + a/G+s + <E>/D+s + u/D+s + <E>/Acc+s + a/Acc+s +  
<E>/Voc+s + <E>/L+s + u/L+s + <E>/I+s + om/I+s;

DJURADJ = :n031s;

DRAGEC = :ARNE + :n021p;

DRAGI = <E>/Nom+s + <B1>o(g/G+s + m/D+s + me/D+s + g/Acc+s + <B1>i/Voc+s + m/L+s +  
me/L+s) + m/I+s + me/I+s;

DUBEC = <E>/Nom+s + <B3>pc(a/G+s + u/D+s + a/Acc+s + <B1>če/Voc+s + u/L+s + em/I+s)  
+ <B3>pc(i/Nom+p + <B2>baca/G+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

EDINBURGH = <E>/Nom+s + a/G+s + u/D+s + <E>/Acc+s + <B1>še/Voc+s + u/L+s + om/I+s;

EGIPAT = <E>/Nom+s + <B2>t(a/G+s + u/D+s) + <E>/Acc+s + <B2>t(e/Voc+s + u/L+s +  
om/I+s);

FA = :FABIANA + :n65p;



FABAC = <E>/Nom+s + <B2>c(a/G+s + u/D+s + a/Acc+s) + <B2>če/Voc+s + <B2>c(u/L+s + em/I+s) + :n361p;

FABBRI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + j(u/L+s + om/I+s) + j(:n021p);

FABEK = :n031s + :n021p;

FABEKOVEC = :ARNE + :n021p;

FABIAN = :n022s + :n021p;

FABIANA = <E>a/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + a/Voc+s + i/L+s + om/I+s);

FABIANO = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s + o/Voc+s + u/L+s + om/I+s);

FABIJANOV = :n022s + :n021p;

FABIK = :n001s + n111p;

FABIO = <E>/Nom+s + <B1>(a/G+s + ja/G+s + ju/D+s + u/D+s + a/Acc+s + ja/Acc+s + o/Voc+s + ju/L+s + u/L+s + jem/I+s + om/I+s);

FABLJAN = :n022s;

FABRETTO = :FABIANO + <B1>(:n021p);

FABRICE = <E>/Nom+s + <B1>(a/G+s + u/D+s + a/Acc+s + e/Voc+s + u/L+s + om/I+s);

FADILJ = :n031s;

FAHRI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + Voc+s + j(u/L+s + om/I+s);

FAHRO = :n50s;

FAIK = :n001s;

FAIVRE = :FABRICE + <B1>(:n021p);

FAKO = :n50s + :APROp;

FALICA = :n5061s + :n65p;

FALKLAND = :n021s;

FANI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s + u/L+s + om/I+s) + <E>/G+s + <E>/D+s + <E>/Acc+s + <E>/Voc+s + <E>/L+s + <E>/I+s;

FANICA = :n5061s;

FARSKI = <E>/Nom+s + <B1>o(g/G+s + m/D+s + me/D+s + g/Acc+s) + <E>/Voc+s + <B1>(om/L+s + ome/L+s + ime/I+s + im/I+s) + <B1>(i/Nom+p + ih/G+p + ima/D+p + e/Acc+p + i/Voc+p + ima/L+p + ima/I+p);

FARSZKY = <E>/Nom+s + <B1>o(ga/G+s + mu/D+s + ga/Acc+s) + <E>/Voc+s + <B1>(ome/L+s + im/I+s) + <B1>(i/Nom+p + yji/Nom+p + ih/G+p + ima/D+p + e/Acc+p + i/Voc+p + yji/Voc+p + ima/L+p + yjima/L+p + ima/I+p + yjima/I+p);

FASLIJA = :AUSTRALIJA + :APROp;

```

FATE = :n5061s;

FAZLI = <E>/Nom+s + j(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s j(u/L+s + om/I+s)
      + j(:n021p);

FELAH = :n0228s;

FELIČE = :ARNE + :n021p;

FEMIJA = <E>/Nom+s + <B1>(e/G+s + i/D+s + u/Acc+s + a/Voc+s + o/Voc+s + i/L+s +
om/I+s);

FENIČANIN = :n022s + <B2>(:n021p);

FERENEC = <E>/Nom+s + <B2>c(a/G+s + u/D+s + a/Acc+s) + <E>/Voc+s + <B2>(če/Voc+s +
cu/L+s + cem/I+s) +
<B2>(ci/Nom+p + aca/G+p + cima/D+p + ce/Acc+p + ci/Voc+p + cima/L+p + cima/I+p);

FERUČO = :FABIANO + <E>/G+s + <E>/D+s + <E>/Acc+s + <E>/L+s + <E>/I+s;

FETAH = :n031s + om/I+s + :n021p;

```

## Prilog 4 – Opis pridjevskih paradigmi (hrCroAdje.nof)

Pozitiv = a/G+m+s+np + u/D+m+s+np + a/Acc+m+s+np + i/Voc+m+s+np + u/L+m+s+np + im/I+m+s+np +

i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np + i/Voc+m+p+np + im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np

+ a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np + oj/L+f+s+np + om/I+f+s+np +

e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np + e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np

+ o/Nom+n+s+np + a/G+n+s+np + u/D+n+s+np + o/Acc+n+s+np + o/Voc+n+s+np + u/L+n+s+np + im/I+n+s+np +

a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np + a/Voc+n+p+np + im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np;

POZvruč = a/G+m+s+np + u/D+m+s+np + a/Acc+m+s+np + i/Voc+m+s+np + u/L+m+s+np + im/I+m+s+np +

i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np + i/Voc+m+p+np + im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np

+ a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np + oj/L+f+s+np + om/I+f+s+np +

e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np + e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np

+ e/Nom+n+s+np + a/G+n+s+np + u/D+n+s+np + e/Acc+n+s+np + e/Voc+n+s+np + u/L+n+s+np + im/I+n+s+np +

a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np + a/Voc+n+p+np + im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np;

KOM = <E>/+cp;

SUP = <E>/+sp;

Komparativ = (i/Nom+m+s + eg/G+m+s + ega/G+m+s + em/D+m+s + emu/D+m+s + eg/Acc+m+s + i/Acc+m+s + ega/Acc+m+s + i/Voc+m+s + em/L+m+s + emu/L+m+s + im/I+m+s +

i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + i/Voc+m+p + im/L+m+p + ima/L+m+p + im/I+m+p + ima/I+m+p)

+ (a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + a/Voc+f+s + oj/L+f+s + om/I+f+s +

a/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + e/Voc+f+p +  
 im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)  
  
 + (e/Nom+n+s + eg/G+n+s + ega/G+n+s + em/D+n+s + emu/D+n+s + e/Acc+n+s +  
 e/Voc+n+s + em/L+n+s + emu/L+n+s + im/I+n+s +  
  
 a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + a/Voc+n+p +  
 im/L+n+p + ima/L+n+p + im/I+n+p + ima/I+n+p);  
  
 MUZIČKI = <E>/Nom+m+s+np + <B1> (og/G+m+s+np + oga/G+m+s+np + om/D+m+s+np +  
 ome/D+m+s+np + omu/D+m+s+np + og/Acc+m+s+np + i/Voc+m+s+np + om/L+m+s+np +  
 ome/L+m+s+np + omu/L+m+s+np + im/I+m+s+np) + <E>/Acc+m+s+np +  
  
 <E>/Nom+m+p+np + h/G+m+p+np + m/D+m+p+np + ma/D+m+p+np + <B1>e/Acc+m+p+np +  
 <E>/Voc+m+p+np + m/L+m+p+np + ma/L+m+p+np + m/I+m+p+np + ma/I+m+p+np  
  
 + <B1>(a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np +  
 oj/L+f+s+np + om/I+f+s+np +  
  
 e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np +  
 e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np)  
  
 + <B1> (o/Nom+n+s+np + og/G+n+s+np + oga/G+n+s+np + ome/D+n+s+np + omu/D+n+s+np  
 + om/D+n+s+np + o/Acc+n+s+np + o/Voc+n+s+np + ome/L+m+s+np + omu/L+m+s+np +  
 om/L+m+s+np + im/I+n+s+np +  
  
 a/Nom+n+p+np) + h/G+n+p+np + m/D+n+p+np + ma/D+n+p+np + <B1>(a/Acc+n+p+np +  
 a/Voc+n+p+np) + m/L+n+p+np + ma/L+n+p+np + m/I+n+p+np + ma/I+n+p+np);  
  
 PRESPOR = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv;  
 PREZREO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B1>l(:Pozitiv);  
 PREŽIVIO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B1>jel(:Pozitiv);  
 PUKAO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>l(:Pozitiv);  
 ŠUPALJ = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>lj(:POZvruč);  
 PREDUGAČAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>k(:Pozitiv);  
 PREKRASAN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>n(:Pozitiv);  
 KADAR1 = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>r(:Pozitiv);  
 MRTAV = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>v(:Pozitiv);  
 CIO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>jel(:Pozitiv);  
 PODSVJESTAN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>n(:Pozitiv);  
 LJUBAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>pk(:Pozitiv);  
 BRIDAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>tk(:Pozitiv);  
 PRETEŽAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>šk(:Pozitiv);  
 DRZAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>sk(:Pozitiv);

ODRASTAO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>l(:Pozitiv);

VRUĆ = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :POZvruć;

LJUDSKI = :MUZIČKI + (j(:Komparativ)):KOM + ((j(:Komparativ))<LW>naj):SUP;

BIJESNI = :MUZIČKI + (j(:Komparativ)<LW><R><S>):KOM +  
 ((j(:Komparativ)<LW><R><S>)<LW>naj):SUP  
 + (<B3>šnj(:Komparativ)<LW><R><S>):KOM +  
 ((<B3>šnj(:Komparativ)<LW><R><S>)<LW>naj):SUP;

UVRIJEĐENI = :MUZIČKI + (j(:Komparativ)<LW><R3><S1>):KOM +  
 ((j(:Komparativ)<LW><R3><S1>)<LW>naj):SUP  
 + (j(:Komparativ)<LW><R3><S2>):KOM +  
 ((j(:Komparativ)<LW><R3><S2>)<LW>naj):SUP;

SVIJETLI = :MUZIČKI + (j(:Komparativ)<LW><R2><S1>):KOM +  
 ((j(:Komparativ)<LW><R2><S1>)<LW>naj):SUP;

PRIJETVORNI = :MUZIČKI + (j(:Komparativ)<LW><R2><S1>):KOM +  
 ((j(:Komparativ)<LW><R2><S1>)<LW>naj):SUP  
 + (j(:Komparativ)<LW><R2><S2>):KOM +  
 ((j(:Komparativ)<LW><R2><S2>)<LW>naj):SUP;

BIJEL = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +  
 (j(:Komparativ)<LW><R1><S1>):KOM + ((j(:Komparativ)<LW><R1><S1>)<LW>naj):SUP;

CIJENJEN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +  
 (ij(:Komparativ)<LW><R1><S1>):KOM + ((ij(:Komparativ)<LW><R1><S1>)<LW>naj):SUP;

SVJEŽ = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (ij(:Komparativ)):KOM +  
 ((ij(:Komparativ))<LW>naj):SUP;

OSVIJETLJEN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +  
 (ij(:Komparativ)<LW><R3><S1>):KOM + ((ij(:Komparativ)<LW><R3><S1>)<LW>naj):SUP;

CRN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (j(:Komparativ)):KOM +  
 ((j(:Komparativ))<LW>naj):SUP;

LIJEN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +  
 (j(:Komparativ)<LW><R1><S1>):KOM + ((j(:Komparativ)<LW><R1><S1>)<LW>naj):SUP;

KRIV = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (lj(:Komparativ)):KOM +  
 (lj(:Komparativ)<LW>naj):SUP;

MEK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (š(:Komparativ)):KOM +  
 ((š(:Komparativ))<LW>naj):SUP;

LIJEP = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +  
 (š(:Komparativ)<LW><R1><S1>):KOM + ((š(:Komparativ)<LW><R1><S1>)<LW>naj):SUP;

JAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>č(:Komparativ)):KOM +  
 ((<B1>č(:Komparativ))<LW>naj):SUP;

```

PRIJEK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +
(<B1>č(:Komparativ)<LW><R2><S1>):KOM +
((<B1>č(:Komparativ)<LW><R2><S1>)<LW>naj):SUP

      + <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +
(<B1>č(:Komparativ)<LW><R2><S2>):KOM +
((<B1>č(:Komparativ)<LW><R2><S2>)<LW>naj):SUP;

ŽUT    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>ć(:Komparativ)):KOM +
((<B1>ć(:Komparativ))<LW>naj):SUP;

MLAD    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>đ(:Komparativ)):KOM +
((<B1>đ(:Komparativ))<LW>naj):SUP;

BLIJED = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv +
(<B1>đ(:Komparativ)<LW><R2><S1>):KOM +
((<B1>đ(:Komparativ)<LW><R2><S1>)<LW>naj):SUP;

PODUG   = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>ž(:Komparativ)):KOM;

SUH      = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>š(:Komparativ)):KOM +
((<B1>š(:Komparativ))<LW>naj):SUP;

DRAG     = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>ž(:Komparativ)):KOM +
((<B1>ž(:Komparativ))<LW>naj):SUP;

BLAG     = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B1>ž(:Komparativ)):KOM +
((<B1>ž(:Komparativ))<LW>naj):SUP;

ŠIROK    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B2>(:Komparativ)):KOM +
((<B2>(:Komparativ))<LW>naj):SUP;

DALEK    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B2>j(:Komparativ)):KOM +
((<B2>j(:Komparativ))<LW>naj):SUP;

DUBOK    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B2>lј(:Komparativ)):KOM +
((<B2>lј(:Komparativ))<LW>naj):SUP;

ČIST     = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B2>šč(:Komparativ)):KOM +
((<B2>šč(:Komparativ))<LW>naj):SUP;

MALEN    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B3>nј(:Komparativ)):KOM +
((<B3>nј(:Komparativ))<LW>naj):SUP;

VELIK    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B3>ć(:Komparativ)):KOM +
((<B3>ć(:Komparativ))<LW>naj):SUP;

VISOK    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv + (<B3>š(:Komparativ)):KOM +
((<B3>š(:Komparativ))<LW>naj):SUP;

DEBEO    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B1>l(:Pozitiv) +
(<B2>lј(:Komparativ)):KOM + ((<B2>lј(:Komparativ))<LW>naj):SUP;

VESEO    = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B1>l(:Pozitiv) +
(<B1>lјј(:Komparativ)):KOM + ((<B1>lјј(:Komparativ))<LW>naj):SUP;

```

DOBAR = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>r(:Pozitiv) +  
 (<BW>bolj(:Komparativ)):KOM + ((<BW>bolj(:Komparativ))<LW>naj):SUP;

BISTAR = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>r(:Pozitiv) +  
 (<B2>rij(:Komparativ)):KOM + ((<B2>rij(:Komparativ))<LW>naj):SUP;

NAGAO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>l(:Pozitiv) +  
 (<B2>lij(:Komparativ)):KOM + ((<B2>lij(:Komparativ))<LW>naj):SUP;

PITAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>k(:Pozitiv) +  
 (<B2>kij(:Komparativ)):KOM + ((<B2>kij(:Komparativ))<LW>naj):SUP;

TANAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>k(:Pozitiv) +  
 (<B2>j(:Komparativ)):KOM + ((<B2>j(:Komparativ))<LW>naj):SUP;

GORAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>k(:Pozitiv) +  
 (<B2>č(:Komparativ)):KOM + ((<B2>č(:Komparativ))<LW>naj):SUP;

TEŽAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>šk:Pozitiv +  
 (<B2>(:Komparativ)):KOM + ((<B2>(:Komparativ))<LW>naj):SUP;

FRIŽAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>šk:Pozitiv +  
 (<B3>šk(:Komparativ)):KOM + ((<B3>šk(:Komparativ))<LW>naj):SUP;

DIVAN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>n(:Pozitiv) +  
 (<B2>nij(:Komparativ)):KOM + ((<B2>nij(:Komparativ))<LW>naj):SUP;

BIJEDAN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>n(:Pozitiv) +  
 (<B2>nij(:Komparativ))<LW><R1><S1>:KOM +  
 ((<B2>nij(:Komparativ))<LW><R1><S1>)<LW>naj):SUP;

KRATAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>k(:Pozitiv) +  
 (<B3>ć(:Komparativ)):KOM + ((<B3>ć(:Komparativ))<LW>naj):SUP;

ZAO = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>l(:Pozitiv) +  
 (<B3>gor(:Komparativ)):KOM + ((<B3>gor(:Komparativ))<LW>naj):SUP;

TIJESAN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B2>n(:Pozitiv) +  
 (<B6>ješnj(:Komparativ)):KOM + ((<B6>ješnj(:Komparativ))<LW>naj):SUP;

NIZAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>sk(:Pozitiv) +  
 (<B3>ž(:Komparativ)):KOM + ((<B3>ž(:Komparativ))<LW>naj):SUP;

KLIZAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>sk(:Pozitiv) +  
 (<B3>sk(:Komparativ)):KOM + ((<B3>sk(:Komparativ))<LW>naj):SUP;

RADOSTAN = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>n(:Pozitiv) +  
 (<B3>nij(:Komparativ)):KOM + ((<B3>nij(:Komparativ))<LW>naj):SUP;

SLADAK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + <B3>tk(:Pozitiv) +  
 (<B3>đ(:Komparativ)):KOM + ((<B3>đ(:Komparativ))<LW>naj):SUP;

LOŠ = :VRUĆ + (ij(:Komparativ)):KOM + ((ij(:Komparativ))<LW>naj):SUP;

GORNJI = <E>/Nom+m+s+np + <B1> (eg/G+m+s+np + ega/G+m+s+np + em/D+m+s+np +  
 emu/D+m+s+np + eg/Acc+m+s+np + ega/Acc+m+s+np + i/Acc+m+s+np + i/Voc+m+s+np +  
 emu/L+m+s+np + em/L+m+s+np + im/I+m+s+np +  
  
 i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np +  
 i/Voc+m+p+np + im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np)  
  
 + <B1> (a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np +  
 oj/L+f+s+np + om/I+f+s+np +  
  
 a/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np +  
 e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np)  
  
 + <B1> (e/Nom+n+s+np + eg/G+n+s+np + ega/G+n+s+np + emu/D+n+s+np + em/D+n+s+np  
 + e/Acc+n+s+np + e/Voc+n+s+np + emu/L+m+s+np + em/L+m+s+np + im/I+n+s+np +  
  
 a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np +  
 a/Voc+n+p+np + im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np);  
  
 ORAHOV = <E>/Nom+m+s+np + a/G+m+s+np + og/G+m+s+np + oga/G+m+s+np + u/D+m+s+np +  
 ome/D+m+s+np + omu/D+m+s+np + om/D+m+s+np + a/Acc+m+s+np + og/Acc+m+s+np +  
 oga/Acc+m+s+np + <E>/Acc+m+s+np + <E>/Voc+m+s+np + u/L+m+s+np + ome/L+m+s+np +  
 omu/L+m+s+np + om/L+m+s+np + im/I+m+s+np +  
  
 i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np +  
 i/Voc+m+p+np + im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np  
  
 + a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np +  
 oj/L+f+s+np + om/I+f+s+np +  
  
 e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np +  
 e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np  
  
 + o/Nom+n+s+np + a/G+n+s+np + og/G+n+s+np + oga/G+n+s+np + u/D+n+s+np +  
 ome/D+n+s+np + omu/D+n+s+np + om/D+n+s+np + o/Acc+n+s+np + o/Voc+n+s+np +  
 u/L+n+s+np + ome/L+m+s+np + omu/L+m+s+np + om/L+m+s+np + im/I+n+s+np +  
  
 a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np +  
 a/Voc+n+p+np + im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np);  
  
 OVELIK = <E>/Nom+m+s+np + <E>/Acc+m+s+np + :Pozitiv  
  
 + <B3>ć(i/Nom+m+s+np + eg/G+m+s+np + ega/G+m+s+np + em/D+m+s+np + emu/D+m+s+np  
 + eg/Acc+m+s+np + ega/Acc+m+s+np + i/Acc+m+s+np + i/Voc+m+s+np + emu/L+m+s+np +  
 em/L+m+s+np + im/I+m+s+np +  
  
 i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np +  
 i/Voc+m+p+np + im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np)  
  
 + <B3>ć(a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + a/Voc+f+s+np +  
 oj/L+f+s+np + om/I+f+s+np +  
  
 e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np +  
 e/Voc+f+p+np + im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np)  
  
 + <B3>ć(e/Nom+n+s+np + eg/G+n+s+np + ega/G+n+s+np + emu/D+n+s+np + em/D+n+s+np  
 + e/Acc+n+s+np + e/Voc+n+s+np + emu/L+m+s+np + em/L+m+s+np + im/I+n+s+np +  
  
 a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np +  
 a/Voc+n+p+np + im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np);



SAM = <E>/Nom+m+s+np + og/G+m+s+np + om/D+m+s+np + og/Acc+m+s+np + om/L+m+s+np +  
im/I+m+s+np + oga/G+m+s+np + ome/D+m+s+np + omu/D+m+s+np + oga/Acc+m+s+np +  
<E>/Acc+m+s+np + ome/L+m+s+np + omu/L+m+s+np + ime/I+m+s+np +

i/Nom+m+p+np + ih/G+m+p+np + im/D+m+p+np + ima/D+m+p+np + e/Acc+m+p+np +  
im/L+m+p+np + ima/L+m+p+np + im/I+m+p+np + ima/I+m+p+np

+ a/Nom+f+s+np + e/G+f+s+np + oj/D+f+s+np + u/Acc+f+s+np + oj/L+f+s+np +  
om/I+f+s+np +

e/Nom+f+p+np + ih/G+f+p+np + im/D+f+p+np + ima/D+f+p+np + e/Acc+f+p+np +  
im/L+f+p+np + ima/L+f+p+np + im/I+f+p+np + ima/I+f+p+np

+ o/Nom+n+s+np + og/G+n+s+np + om/D+n+s+np + og/Acc+n+s+np + om/L+n+s+np +  
im/I+n+s+np + oga/G+n+s+np + ome/D+n+s+np + omu/D+n+s+np + oga/Acc+n+s+np +  
ome/L+n+s+np + omu/L+n+s+np + ime/I+n+s+np +

a/Nom+n+p+np + ih/G+n+p+np + im/D+n+p+np + ima/D+n+p+np + a/Acc+n+p+np +  
im/L+n+p+np + ima/L+n+p+np + im/I+n+p+np + ima/I+n+p+np;

## Prilog 5 – Opis glagolskih paradigmi (hrCroVerb.nof)

infinitiv = <E>/INF + <Bl>/INFk;

IMP = h/IMP+s+m+1 + h/IMP+s+f+1 + h/IMP+s+n+1 + še/IMP+s+m+2 + še/IMP+s+f+2 +  
še/IMP+s+n+2 + še/IMP+s+m+3 + še/IMP+s+f+3 + še/IMP+s+n+3

+ smo/IMP+p+m+1 + smo/IMP+p+f+1 + smo/IMP+p+n+1 + ste/IMP+p+m+2 + ste/IMP+p+f+2 +  
ste/IMP+p+n+2 + hu/IMP+p+m+3 + hu/IMP+p+f+3 + hu/IMP+p+n+3;

IMT = m/IMT+s+m+1 + <E>/IMT+s+m+2 + <E>/IMT+s+m+3 + mo/IMT+p+m+1 + te/IMT+p+m+2 +  
<Bl>e/IMT+p+m+3 +

m/IMT+s+f+1 + <E>/IMT+s+f+2 + <E>/IMT+s+f+3 + mo/IMT+p+f+1 + te/IMT+p+f+2 +  
<Bl>e/IMT+p+f+3 +

m/IMT+s+n+1 + <E>/IMT+s+n+2 + <E>/IMT+s+n+3 + mo/IMT+p+n+1 + te/IMT+p+n+2 +  
<Bl>e/IMT+p+n+3;

IMTbirati = <B2>(m/IMT+s+m+1 + j/IMT+s+m+2 + <E>/IMT+s+m+3 + jmo/IMT+p+m+1 +  
jte/IMT+p+m+2 + ju/IMT+p+m+3 +

m/IMT+s+f+1 + j/IMT+s+f+2 + <E>/IMT+s+f+3 + jmo/IMT+p+f+1 +  
jte/IMT+p+f+2 + ju/IMT+p+f+3 +

m/IMT+s+n+1 + j/IMT+s+n+2 + <E>/IMT+s+n+3 + jmo/IMT+p+n+1 +  
jte/IMT+p+n+2 + ju/IMT+p+n+3);

IMTbljesnuti = <B3>(em/IMT+s+m+1 + e/IMT+s+m+2 + e/IMT+s+m+3 + imo/IMT+p+m+1 +  
ite/IMT+p+m+2 + u/IMT+p+m+3 +

em/IMT+s+f+1 + e/IMT+s+f+2 + e/IMT+s+f+3 + imo/IMT+p+f+1 +  
ite/IMT+p+f+2 + u/IMT+p+f+3 +

em/IMT+s+n+1 + e/IMT+s+n+2 + e/IMT+s+n+3 + imo/IMT+p+n+1 +  
ite/IMT+p+n+2 + u/IMT+p+n+3);

IMT2 = em/IMT+s+m+1 + <E>/IMT+s+m+2 + e/IMT+s+m+3 + mo/IMT+p+m+1 + te/IMT+p+m+2 +  
u/IMT+p+m+3 +

em/IMT+s+f+1 + <E>/IMT+s+f+2 + e/IMT+s+f+3 + mo/IMT+p+f+1 + te/IMT+p+f+2 +  
u/IMT+p+f+3 +

em/IMT+s+n+1 + <E>/IMT+s+n+2 + e/IMT+s+n+3 + mo/IMT+p+n+1 + te/IMT+p+n+2 +  
u/IMT+p+n+3;

IMT3 = em/IMT+s+m+1 + i/IMT+s+m+2 + e/IMT+s+m+3 + imo/IMT+p+m+1 + ite/IMT+p+m+2 +  
u/IMT+p+m+3 +

em/IMT+s+f+1 + i/IMT+s+f+2 + e/IMT+s+f+3 + imo/IMT+p+f+1 + ite/IMT+p+f+2 +  
u/IMT+p+f+3 +

em/IMT+s+n+1 + i/IMT+s+n+2 + e/IMT+s+n+3 + imo/IMT+p+n+1 + ite/IMT+p+n+2 +  
u/IMT+p+n+3;

$$\begin{aligned} \text{IMTdati} = & \langle B2 \rangle (m/\text{IMT}+s+m+1 + \langle E \rangle / \text{IMT}+s+m+3 + \langle B1 \rangle u/\text{IMT}+p+m+3 + \\ & m/\text{IMT}+s+f+1 + \langle E \rangle / \text{IMT}+s+f+3 + \langle B1 \rangle u/\text{IMT}+p+f+3 + \\ & m/\text{IMT}+s+n+1 + \langle E \rangle / \text{IMT}+s+n+3 + \langle B1 \rangle u/\text{IMT}+p+n+3); \\ \text{IMTisteći} = & \langle B2 \rangle (\check{c}em/\text{IMT}+s+m+1 + ci/\text{IMT}+s+m+2 + \check{c}e/\text{IMT}+s+m+3 + cimo/\text{IMT}+p+m+1 + \\ & cite/\text{IMT}+p+m+2 + ku/\text{IMT}+p+m+3 + \\ & \check{c}em/\text{IMT}+s+f+1 + ci/\text{IMT}+s+f+2 + \check{c}e/\text{IMT}+s+f+3 + cimo/\text{IMT}+p+f+1 + \\ & cite/\text{IMT}+p+f+2 + ku/\text{IMT}+p+f+3 + \\ & \check{c}em/\text{IMT}+s+n+1 + ci/\text{IMT}+s+n+2 + \check{c}e/\text{IMT}+s+n+3 + cimo/\text{IMT}+p+n+1 + \\ & cite/\text{IMT}+p+n+2 + ku/\text{IMT}+p+n+3); \\ \text{IMTizleći} = & \langle B2 \rangle (\check{z}em/\text{IMT}+s+m+1 + zi/\text{IMT}+s+m+2 + \check{z}e/\text{IMT}+s+m+3 + zimo/\text{IMT}+p+m+1 + \\ & zite/\text{IMT}+p+m+2 + gu/\text{IMT}+p+m+3 + \\ & \check{z}em/\text{IMT}+s+f+1 + zi/\text{IMT}+s+f+2 + \check{z}e/\text{IMT}+s+f+3 + zimo/\text{IMT}+p+f+1 + \\ & zite/\text{IMT}+p+f+2 + gu/\text{IMT}+p+f+3 + \\ & \check{z}em/\text{IMT}+s+n+1 + zi/\text{IMT}+s+n+2 + \check{z}e/\text{IMT}+s+n+3 + zimo/\text{IMT}+p+n+1 + \\ & zite/\text{IMT}+p+n+2 + gu/\text{IMT}+p+n+3); \\ \text{AO1} = & h/\text{AO}+s+m+1 + \langle E \rangle / \text{AO}+s+m+2 + \langle E \rangle / \text{AO}+s+m+3 + smo/\text{AO}+p+m+1 + ste/\text{AO}+p+m+2 + \\ & \check{s}e/\text{AO}+p+m+3 + \\ & h/\text{AO}+s+f+1 + \langle E \rangle / \text{AO}+s+f+2 + \langle E \rangle / \text{AO}+s+f+3 + smo/\text{AO}+p+f+1 + ste/\text{AO}+p+f+2 + \\ & \check{s}e/\text{AO}+p+f+3 + \\ & h/\text{AO}+s+n+1 + \langle E \rangle / \text{AO}+s+n+2 + \langle E \rangle / \text{AO}+s+n+3 + smo/\text{AO}+p+n+1 + ste/\text{AO}+p+n+2 + \\ & \check{s}e/\text{AO}+p+n+3; \\ \text{AO2} = & oh/\text{AO}+s+m+1 + e/\text{AO}+s+m+2 + e/\text{AO}+s+m+3 + osmo/\text{AO}+p+m+1 + oste/\text{AO}+p+m+2 + \\ & o\check{s}e/\text{AO}+p+m+3 + \\ & oh/\text{AO}+s+f+1 + e/\text{AO}+s+f+2 + e/\text{AO}+s+f+3 + osmo/\text{AO}+p+f+1 + oste/\text{AO}+p+f+2 + \\ & o\check{s}e/\text{AO}+p+f+3 + \\ & oh/\text{AO}+s+n+1 + e/\text{AO}+s+n+2 + e/\text{AO}+s+n+3 + osmo/\text{AO}+p+n+1 + oste/\text{AO}+p+n+2 + \\ & o\check{s}e/\text{AO}+p+n+3; \\ \text{AOg} = & goh/\text{AO}+s+m+1 + \check{z}e/\text{AO}+s+m+2 + \check{z}e/\text{AO}+s+m+3 + gosmo/\text{AO}+p+m+1 + goste/\text{AO}+p+m+2 + \\ & go\check{s}e/\text{AO}+p+m+3 + \\ & goh/\text{AO}+s+f+1 + \check{z}e/\text{AO}+s+f+2 + \check{z}e/\text{AO}+s+f+3 + gosmo/\text{AO}+p+f+1 + goste/\text{AO}+p+f+2 + \\ & go\check{s}e/\text{AO}+p+f+3 + \\ & goh/\text{AO}+s+n+1 + \check{z}e/\text{AO}+s+n+2 + \check{z}e/\text{AO}+s+n+3 + gosmo/\text{AO}+p+n+1 + goste/\text{AO}+p+n+2 + \\ & go\check{s}e/\text{AO}+p+n+3; \\ \text{AOk} = & koh/\text{AO}+s+m+1 + \check{c}e/\text{AO}+s+m+2 + \check{c}e/\text{AO}+s+m+3 + kosmo/\text{AO}+p+m+1 + koste/\text{AO}+p+m+2 + \\ & ko\check{s}e/\text{AO}+p+m+3 + \\ & koh/\text{AO}+s+f+1 + \check{c}e/\text{AO}+s+f+2 + \check{c}e/\text{AO}+s+f+3 + kosmo/\text{AO}+p+f+1 + koste/\text{AO}+p+f+2 + \\ & ko\check{s}e/\text{AO}+p+f+3 + \\ & koh/\text{AO}+s+n+1 + \check{c}e/\text{AO}+s+n+2 + \check{c}e/\text{AO}+s+n+3 + kosmo/\text{AO}+p+n+1 + koste/\text{AO}+p+n+2 + \\ & ko\check{s}e/\text{AO}+p+n+3; \\ \text{AO2biti} = & \langle B2 \rangle (h/\text{AO2}+s+m+1 + \langle E \rangle / \text{AO2}+s+m+2 + \langle E \rangle / \text{AO2}+s+m+3 + smo/\text{AO2}+p+m+1 + \\ & ste/\text{AO2}+p+m+2 + \langle E \rangle / \text{AO2}+p+m+3) +
\end{aligned}$$

```

    <B2>(h/AO2+s+f+1 + <E>/AO2+s+f+2 + <E>/AO2+s+f+3 + smo/AO2+p+f+1 +
ste/AO2+p+f+2 + <E>/AO2+p+f+3) +

    <B2>(h/AO2+s+n+1 + <E>/AO2+s+n+2 + <E>/AO2+s+n+3 + smo/AO2+p+n+1 +
ste/AO2+p+n+2 + <E>/AO2+p+n+3);

PDR = o/PDR+s+1+m + o/PDR+s+2+m + o/PDR+s+3+m + la/PDR+s+1+f + la/PDR+s+2+f +
la/PDR+s+3+f + lo/PDR+s+1+n + lo/PDR+s+2+n + lo/PDR+s+3+n

    + li/PDR+p+1+m + li/PDR+p+2+m + li/PDR+p+3+m + le/PDR+p+1+f + le/PDR+p+2+f +
le/PDR+p+3+f + la/PDR+p+1+n + la/PDR+p+2+n + la/PDR+p+3+n;

PDR2 = ao/PDR+s+1+m + ao/PDR+s+2+m + ao/PDR+s+3+m + la/PDR+s+1+f + la/PDR+s+2+f +
la/PDR+s+3+f + lo/PDR+s+1+n + lo/PDR+s+2+n + lo/PDR+s+3+n

    + li/PDR+p+1+m + li/PDR+p+2+m + li/PDR+p+3+m + le/PDR+p+1+f + le/PDR+p+2+f +
le/PDR+p+3+f + la/PDR+p+1+n + la/PDR+p+2+n + la/PDR+p+3+n;

PDRdoživjeti = <B4>i(o/PDR+s+1+m + o/PDR+s+2+m + o/PDR+s+3+m) + <B2>(la/PDR+s+1+f +
la/PDR+s+2+f + la/PDR+s+3+f + lo/PDR+s+1+n + lo/PDR+s+2+n + lo/PDR+s+3+n

    + li/PDR+p+1+m + li/PDR+p+2+m + li/PDR+p+3+m + le/PDR+p+1+f +
le/PDR+p+2+f + le/PDR+p+3+f + la/PDR+p+1+n + la/PDR+p+2+n + la/PDR+p+3+n);

PDT = <E>/PDT+s+1+m + a/PDT+s+1+f + o/PDT+s+1+n + i/PDT+p+1+m + e/PDT+p+1+f +
a/PDT+p+1+n +

    <E>/PDT+s+2+m + a/PDT+s+2+f + o/PDT+s+2+n + i/PDT+p+2+m + e/PDT+p+2+f +
a/PDT+p+2+n +

    <E>/PDT+s+3+m + a/PDT+s+3+f + o/PDT+s+3+n + i/PDT+p+3+m + e/PDT+p+3+f +
a/PDT+p+3+n;

PRu = (m/PR+s+m+1 + m/PR+s+f+1 + m/PR+s+n+1 + š/PR+s+m+2 + š/PR+s+f+2 + š/PR+s+n+2
+ <E>/PR+s+m+3 + <E>/PR+s+f+3 + <E>/PR+s+n+3 + mo/PR+p+m+1 + mo/PR+p+f+1 +
mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2 + te/PR+p+n+2 + <B1>u/PR+p+m+3 +
<B1>u/PR+p+f+3 + <B1>u/PR+p+n+3);

PRe = (m/PR+s+m+1 + m/PR+s+f+1 + m/PR+s+n+1 + š/PR+s+m+2 + š/PR+s+f+2 + š/PR+s+n+2
+ <E>/PR+s+m+3 + <E>/PR+s+f+3 + <E>/PR+s+n+3 + mo/PR+p+m+1 + mo/PR+p+f+1 +
mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2 + te/PR+p+n+2 + <B1>e/PR+p+m+3 +
<B1>e/PR+p+f+3 + <B1>e/PR+p+n+3);

PRbirati = <B2>(m/PR+s+m+1 + m/PR+s+f+1 + m/PR+s+n+1 + š/PR+s+m+2 + š/PR+s+f+2 +
š/PR+s+n+2 + <E>/PR+s+m+3 + <E>/PR+s+f+3 + <E>/PR+s+n+3

    + mo/PR+p+m+1 + mo/PR+p+f+1 + mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2 +
te/PR+p+n+2 + ju/PR+p+m+3 + ju/PR+p+f+3 + ju/PR+p+n+3);

PRisteći = <B2>če(m/PR+s+m+1 + m/PR+s+f+1 + m/PR+s+n+1 + š/PR+s+m+2 + š/PR+s+f+2 +
š/PR+s+n+2 + <E>/PR+s+m+3 + <E>/PR+s+f+3 + <E>/PR+s+n+3

    + mo/PR+p+m+1 + mo/PR+p+f+1 + mo/PR+p+n+1 + te/PR+p+m+2 +
te/PR+p+f+2 + te/PR+p+n+2 + <B2>(ku/PR+p+m+3 + ku/PR+p+f+3 + ku/PR+p+n+3));

PRizleći = <B2>že(m/PR+s+m+1 + m/PR+s+f+1 + m/PR+s+n+1 + š/PR+s+m+2 + š/PR+s+f+2 +
š/PR+s+n+2 + <E>/PR+s+m+3 + <E>/PR+s+f+3 + <E>/PR+s+n+3

    + mo/PR+p+m+1 + mo/PR+p+f+1 + mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2
+ te/PR+p+n+2) + <B2>(gu/PR+p+m+3 + gu/PR+p+f+3 + gu/PR+p+n+3));

PREDATI = :DODATI + <B2>de(:PRu) + <B2>dne(:PRu);

```

```

BITI = :infinitiv

      + <B4>jes (am/PR+s+m+1 + am/PR+s+f+1 + am/PR+s+n+1 + i/PR+s+m+2 + i/PR+s+f+2
+ i/PR+s+n+2 + t/PR+s+m+3 + t/PR+s+f+3 + t/PR+s+n+3 + mo/PR+p+m+1 + mo/PR+p+f+1 +
mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2 + te/PR+p+n+2 + u/PR+p+m+3 + u/PR+p+f+3 +
u/PR+p+n+3)

      + <B4>s (am/PRk+s+m+1 + am/PRk+s+f+1 + am/PRk+s+n+1 + i/PRk+s+m+2 +
i/PRk+s+f+2 + i/PRk+s+n+2 + <B1>(je/PRk+s+m+3 + je/PRk+s+f+3 + je/PRk+s+n+3) +
mo/PRk+p+m+1 + mo/PRk+p+f+1 + mo/PRk+p+n+1 + te/PRk+p+m+2 + te/PRk+p+f+2 +
te/PRk+p+n+2 + u/PRk+p+m+3 + u/PRk+p+f+3 + u/PRk+p+n+3)

      + <B3>ude (m/PRd+s+m+1 + š/PRd+s+m+2 + <E>/PRd+s+m+3 + mo/PRd+p+m+1 +
te/PRd+p+m+2 + <B1>u/PRd+p+m+3

      + m/PRd+s+f+1 + š/PRd+s+f+2 + <E>/PRd+s+f+3 + mo/PRd+p+f+1 +
te/PRd+p+f+2 + <B1>u/PRd+p+f+3

      + m/PRd+s+n+1 + š/PRd+s+n+2 + <E>/PRd+s+n+3 + mo/PRd+p+n+1 +
te/PRd+p+n+2 + <B1>u/PRd+p+n+3)

      + <B2>ja(:IMP)

      + <B3>je (h/IMP2+s+m+1 + še/IMP2+s+m+2 + še/IMP2+s+m+3 + smo/IMP2+p+m+1 +
ste/IMP2+p+m+2 + hu/IMP2+p+m+3

      + h/IMP2+s+f+1 + še/IMP2+s+f+2 + še/IMP2+s+f+3 + smo/IMP2+p+f+1 +
ste/IMP2+p+f+2 + hu/IMP2+p+f+3

      + h/IMP2+s+n+1 + še/IMP2+s+n+2 + še/IMP2+s+n+3 + smo/IMP2+p+n+1 +
ste/IMP2+p+n+2 + hu/IMP2+p+n+3)

      + <B3>je (:AO1) + :AO2biti + <B2>(:PDR)

      + <B3>ud (em/IMT+s+m+1 + e/IMT+s+m+2 + e/IMT+s+m+3 + imo/IMT+p+m+1 +
ite/IMT+p+m+2 + u/IMT+p+m+3 +

      em/IMT+s+f+1 + e/IMT+s+f+2 + e/IMT+s+f+3 + imo/IMT+p+f+1 +
ite/IMT+p+f+2 + u/IMT+p+f+3 +

      em/IMT+s+n+1 + e/IMT+s+n+2 + e/IMT+s+n+3 + imo/IMT+p+n+1 +
ite/IMT+p+n+2 + u/IMT+p+n+3)

      + <B3>udući/GPS + <B2>vši/GPP;

BAVITI = :infinitiv + <B2>(:Pre) + <B3>lja(:IMP) + <B2>(:PDR) + <B2>(:IMT) +
<B3>eći/GPS;

BILJEŽITI = :infinitiv + <B2>(:Pre) + <B3>a(:IMP) + <B2>(:PDR) + <B3>en(:PDT) +
<B2>(:IMT) + <B3>eći/GPS;

BIRATI = :infinitiv + :PRbirati + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
:IMTbirati + <B2>jući/GPS;

BLJESNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1) + <B2>(:PDR) + :IMTbljesnuti +
<B2>vši/GPP;

BOJATI = :infinitiv + <B3>i(:Pre) + <B2>(:IMP) + <B2>(:PDR) + <B3>i(:IMT) +
<B3>eći/GPS;

BACITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B4>čen(:PDT) +
<B2>(:IMT) + <B2>vši/GPP;

```

```

BRANITI = :infinitiv + <B2>(:Pre) + <B3>ja(:IMP) + <B2>(:PDR) + <B2>(:IMT) +
<B3>eći/GPS;

DAROVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +
<B5>uj(:IMT2) + <B2>vši/GPP;

BROJATI = :infinitiv + <B3>i(:Pre) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
B3>i(:IMT) + <B3>eći/GPS;

CAREVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:PDR) + <B2>(:IMP) + <B5>uj(:IMT2) +
<B5>ujući/GPS;

CIJEPIITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B3>ljen(:PDT) +
<B2>(:IMT) + <B2>vši/GPP;

CRPSTI = :infinitiv + <B3>e(:PRu) + <B2>(:PDR2) + <B3>ija(:IMP) + <B3>en(:PDT) +
<B3>(:IMT3) + <B3>ući/GPS;

CVILITI = :infinitiv + <B2>(:Pre) + <B3>ja(:IMP) + <B2>(:PDR) + <B2>(:IMT) +
<B3>eći/GPS;

ČUDITI = :infinitiv + <B2>(:Pre) + <B4>đa(:IMP) + <B2>(:PDR) + <B2>(:IMT) +
<B3>eći/GPS;

ČUTI = :infinitiv + <B2>je(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>ven(:PDT) +
<B1>(:PDT) + <B2>j(:IMT2) + <B2>vši/GPP;

DATI = :infinitiv + :PRbirati + <B2>de(:PRu) + <B2>dne(:PRu)

      + <B2>do (h/AO+s+m+1 + <B1>e/AO+s+m+2 + <B1>e/AO+s+m+3 + smo/AO+p+m+1 +
ste/AO+p+m+2 + še/AO+p+m+3 +

      h/AO+s+f+1 + <B1>e/AO+s+f+2 + <B1>e/AO+s+f+3 + smo/AO+p+f+1 +
ste/AO+p+f+2 + še/AO+p+f+3 +

      h/AO+s+n+1 + <B1>e/AO+s+n+2 + <B1>e/AO+s+n+3 + smo/AO+p+n+1 +
ste/AO+p+n+2 + še/AO+p+n+3)

      + <B2>(:PDR) + <B2>n(:PDT) + :IMTbirati + :IMTdatil

      + <B2>dne (m/IMT+s+m+1 + <E>/IMT+s+m+3 + <B1>u/IMT+p+m+3 +

      m/IMT+s+f+1 + <E>/IMT+s+f+3 + <B1>u/IMT+p+f+3 +

      m/IMT+s+n+1 + <E>/IMT+s+n+3 + <B1>u/IMT+p+n+3)

      + <B2>vši/GPP;

DIĆI = <E>/INF + <E>/INFk + <B2>gne(:PRu) + <B2>(:AOg) + <B2>g(:PDR2) +
<B2>gnut(:PDT) + <B2>gn(:IMT3) + <B2>gavši/GPP;

DOĆI = <E>/INF + <E>/INFk + <B2>đe (:PRu) + <B2>d(:AO2) + <B2>š (:PDR2) + <B2>d
(:IMT3) + <B2>šavši/GPP;

DOLAZITI = :infinitiv + <B2>(:Pre) + <B3>a(:IMP) + <B2>(:PDR) + <B2>(:IMT) +
<B3>eći/GPS;

HTJETI = :infinitiv

      + <B5>oć(u/PR+s+m+1 + u/PR+s+f+1 + u/PR+s+n+1 + eš/PR+s+m+2 + eš/PR+s+f+2 +
eš/PR+s+n+2 + e/PR+s+m+3 + e/PR+s+f+3 + e/PR+s+n+3

      + emo/PR+p+m+1 + emo/PR+p+f+1 + emo/PR+p+n+1 + ete/PR+p+m+2 +
ete/PR+p+f+2 + ete/PR+p+n+2 + e/PR+p+m+3 + e/PR+p+f+3 + e/PR+p+n+3)

```

```

+ <B6>ć (u/PR+s+m+1 + u/PR+s+f+1 + u/PR+s+n+1 + eš/PR+s+m+2 + eš/PR+s+f+2 +
eš/PR+s+n+2 + e/PR+s+m+3 + e/PR+s+f+3 + e/PR+s+n+3

+ emo/PR+p+m+1 + emo/PR+p+f+1 + emo/PR+p+n+1 + ete/PR+p+m+2 +
ete/PR+p+f+2 + ete/PR+p+n+2 + e/PR+p+m+3 + e/PR+p+f+3 + e/PR+p+n+3)

+ <B2>dne(:PRu) + <B4>ija(:IMP)

+ <B5>otija (h/IMP2+s+1 + še/IMP2+s+2 + še/IMP2+s+3 + smo/IMP2+p+1 +
ste/IMP2+p+2 + hu/IMP2+p+3)

+ <B5>oća (h/IMP3+s+1 + še/IMP3+s+2 + še/IMP3+s+3 + smo/IMP3+p+1 +
ste/IMP3+p+2 + hu/IMP3+p+3)

+ <B2>d(:AO2) + :AO2biti + :PDRdoživjeti + <B2>dn(:IMT3) + <B5>oteći/GPS +
<B2>vši/GPP + <B5>otjevši/GPP;

DODATI = :infinitiv + :PRbirati + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +
:IMTbirati + <B2>vši/GPP;

DODAVATI = :infinitiv + <B4>je(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
<B4>j(:IMT2) + <B4>jući/GPS;

DOTAĆI = :infinitiv + <B2>kne(:PRu) + <B2>(:AOk) + <B2>k(:PDR2) + :IMTisteći +
<B2>kavši/GPP;

DOŽIVJETI = :infinitiv + <B4>i(:Pre) + :PDRdoživjeti + <B2>(:AO1) + <B4>ljen(:PDT)
+ <B4>i(:IMT) + <B2>vši/GPP;

DRŽATI = :infinitiv + <B3>i(:Pre) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
B3>i(:IMT) + <B3>eći/GPS;

GOSTITI = :infinitiv + <B2>(:Pre) + <B5>šća(:IMP) + <B2>(:PDR) + <B5>šćen(:PDT) +
<B2>(:IMT) + <B3>eći/GPS;

IĆI = :infinitiv + <B2>de(:PRu) + <B2>đa(:IMP) + <B2>š(:PDR2) + <B2>d (:IMT3) +
<B2>dući/GPS;

IZABRATI = :infinitiv + <B4>ere(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +
<B4>er(:IMT3) + <B2>vši/GPP;

IZLAGATI = :infinitiv + <B4>že(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
<B4>ž(:IMT3) + <B4>žući/GPS;

IZVADITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B4>đen(:PDT) +
<B2>(:IMT) + <B2>vši/GPP;

IZVOJEVATI = :BIRATI + <B2>vši/GPS;

JAHATI = :infinitiv + <B4>še (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B4>š
(:IMT3) + <B4>šući/GPS;

JESTI = :infinitiv + <B3>de(:PRu) + <B3>diја(:IMP) + <B3>(:PDR) + <B3>den(:PDT) +
<B3>d(:IMT3) + <B3>dući/GPS;

KAZATI = :infinitiv + <B4>že (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +
<B4>ž(:IMT3) + <B2>vši/GPP;

KAZNITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B5>žnjen(:PDT) +
<B2>(:IMT) + <B2>vši/GPP;

KLICATI = :infinitiv + <B4>če (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>č(:IMT3) +
<B4>čući/GPS;

```

KORISTITI = :infinitiv + <B2>(:Pre) + <B5>šća(:IMP) + <B2>(:PDR) + <B5>šten(:PDT) + <B2>(:IMT) + <B3>eći/GPS;

KOSITI = :infinitiv + <B2>(:Pre) + <B4>ša(:IMP) + <B2>(:PDR) + <B4>šen(:PDT) + <B2>(:IMT) + <B3>eći/GPS;

KRETATI = :infinitiv + <B4>će (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>ć (:IMT3) + <B4>ćući/GPS;

LJUTITI = :infinitiv + <B2>(:Pre) + <B4>ća(:IMP) + <B2>(:PDR) + <B4>ćen(:PDT) + <B2>(:IMT) + <B3>eći/GPS;

MICATI = :infinitiv + <B4>če (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B4>č(:IMT3) + <B4>ćući/GPS;

MIRISATI = :infinitiv + <B4>še (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>š (:IMT3) + <B4>šući/GPS;

MISLITI = :infinitiv + <B2>(:Pre) + <B5>šlja(:IMP) + <B2>(:PDR) + <B5>šljen(:PDT) + <B2>(:IMT) + <B3>eći/GPS;

MOĆI = :infinitiv + <B2>gne (:PRu)

+ <B2> (goh/IMP+s+m+1 + že/IMP+s+m+2 + že/IMP+s+m+3 + gosmo/IMP+p+m+1 + goste/IMP+p+m+2 + gaše/IMP+p+m+3 +

goh/IMP+s+f+1 + že/IMP+s+f+2 + že/IMP+s+f+3 + gosmo/IMP+p+f+1 + goste/IMP+p+f+2 + gaše/IMP+p+f+3 +

goh/IMP+s+n+1 + že/IMP+s+n+2 + že/IMP+s+n+3 + gosmo/IMP+p+n+1 + goste/IMP+p+n+2 + gaše/IMP+p+n+3)

+ <B2>g(:PDR2) + <B2>gn(:IMT3) + <B2>gavši/GPP

+ <B2> (gu/PR+s+m+1 + gu/PR+s+f+1 + gu/PR+s+n+1 + žeš/PR+s+m+2 + žeš/PR+s+f+2 + žeš/PR+s+n+2 + že/PR+s+m+3 + že/PR+s+f+3 + že/PR+s+n+3 + žemo/PR+p+m+1 + žemo/PR+p+f+1 + žemo/PR+p+n+1 + žete/PR+p+m+2 + žete/PR+p+f+2 + žete/PR+p+n+2 + gu/PR+p+m+3 + gu/PR+p+f+3 + gu/PR+p+n+3)

+ <B2>ga(:IMP)

+ <B2> (gu/IMT+s+m+1 + zi/IMT+s+m+2 + že/IMT+s+m+3 + zimo/IMT+p+m+1 + zite/IMT+p+m+2 + gu/IMT+p+m+3) + <B2>gni (<E>/IMT+s+m+2 + mo/IMT+p+m+1 + te/IMT+p+m+2) +

<B2> (gu/IMT+s+f+1 + zi/IMT+s+f+2 + že/IMT+s+f+3 + zimo/IMT+p+f+1 + zite/IMT+p+f+2 + gu/IMT+p+f+3) + <B2>gni (<E>/IMT+s+f+2 + mo/IMT+p+f+1 + te/IMT+p+f+2) +

<B2> (gu/IMT+s+n+1 + zi/IMT+s+n+2 + že/IMT+s+n+3 + zimo/IMT+p+n+1 + zite/IMT+p+n+2 + gu/IMT+p+n+3) + <B2>gni (<E>/IMT+s+n+2 + mo/IMT+p+n+1 + te/IMT+p+n+2)

+ <B2>gući/GPs;

NAĆI = :infinitiv + <B2>đe(:PRu) + <B2>đ(:AO2) + <B2>š(:PDR2) + <B2>đen(:PDT) + <B2>đ(:IMT3) + <B2>šavši/GPP;

NAGNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B1>(:PDT) + <B3>(:IMT3) + <B2>vši/GPP;

NAMETNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B1>(:PDT) + <B3>(:IMT3) + <B2>vši/GPP;



NAPATITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B4>ćen(:PDT) + <B2>(:IMT) + <B2>vši/GPP;

NAPITI = :infinitiv + <B2>je(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>j(:IMT2) + <B2>vši/GPP;

NARASTI = :infinitiv + <B1>e(:PRu) + <B1>(:AO2) + <B2>(:PDR2) + <B1>(:IMT3) + <B1>avši/GPP;

NARUČIVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B5>uj(:IMT2) + <B5>ujući/GPS;

NASMIJATI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) + <B3>(:IMT2) + <B2>vši/GPP;

NEĆU = <E>/PR+s+m+1 + <E>/PR+s+f+1 + <E>/PR+s+n+1 + <B1>e(š/PR+s+m+2 + š/PR+s+f+2 + š/PR+s+n+2 + <E>/PR+s+m+3 + <E>/PR+s+f+3 + <E>/PR+s+n+3 + mo/PR+p+m+1 + mo/PR+p+f+1 + mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2 + te/PR+p+n+2 + <E>/PR+p+m+3 + <E>/PR+p+f+3 + <E>/PR+p+n+3)

+ <E>/IMT+s+m+1 + <B1>(e/IMT+s+m+3 + e/IMT+p+m+3) +

<E>/IMT+s+f+1 + <B1>(e/IMT+s+f+3 + e/IMT+p+f+3) +

<E>/IMT+s+n+1 + <B1>(e/IMT+s+n+3 + e/IMT+p+n+3);

NISAM = <E>/PR+s+m+1 + <E>/PR+s+f+1 + <E>/PR+s+n+1 + <B2>(i/PR+s+m+2 + i/PR+s+f+2 + i/PR+s+n+2 + <B1>je/PR+s+m+3 + je/PR+s+f+3 + je/PR+s+n+3 + mo/PR+p+m+1 + mo/PR+p+f+1 + mo/PR+p+n+1 + te/PR+p+m+2 + te/PR+p+f+2 + te/PR+p+n+2 + u/PR+p+m+3 + u/PR+p+f+3 + u/PR+p+n+3)

+ <E>/IMT+s+m+1 + <B3>je/IMT+s+m+3 + <B2>u/IMT+p+m+3 +

<E>/IMT+s+f+1 + <B3>je/IMT+s+f+3 + <B2>u/IMT+p+f+3 +

<E>/IMT+s+n+1 + <B3>je/IMT+s+n+3 + <B2>u/IMT+p+n+3;

NJIHATI = :infinitiv + <B4>še(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>š(:IMT3) + <B4>šući/GPS;

OBJASNITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B5>šnjjen(:PDT) + <B2>(:IMT) + <B2>vši/GPP;

ODŠETATI = :infinitiv + :PRbirati + <B2>(:AO1) + <B2>(:PDR) + <B4>ć(:IMT3) + <B2>vši/GPP;

ODMICATI = :infinitiv + <B4>če(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B4>č(:IMT3) + <B4>čući/GPS;

OKRETATI = :infinitiv + <B4>će(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B4>ć(:IMT3) + <B4>čući/GPS;

OMEŠTI = :infinitiv + <B3>te(:PRu) + <B3>t(:AO2) + <B3>(:PDR) + <B3>ten(:PDT) + <B3>t(:IMT3) + <B3>(vši/GPP + tavši/GPP);

OPIRATI = :infinitiv + <B3>e(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B3>(:IMT3) + <B3>ući/GPS;

OTPUTOVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B5>uj(:IMT2) + <B2>vši/GPP;

OVISITI = :infinitiv + <B2>(:Pre) + <B4>ša(:IMP) + <B2>(:PDR) + <B2>(:IMT) + <B3>eći/GPS;

PATENTIRATI = :infinitiv + :PRbirati + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + :IMTbirati + <B2>jući/GPS;

PITI = :infinitiv + <B2>je(:PRu) + <B2>ja(:IMP) + <B2>(:PDR) + <B2>jen(:PDT) + <B1>(:PDT) + <B2>j(:IMT2) + <B2>jući/GPS;

PISATI = :infinitiv + <B4>še (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B4>š(:IMT3) + <B4>ući/GPS;

PLJESKATI = <E>/INF + <B5>šće(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B5>šč(:IMT3) + <B5>ščući/GPS;

POKLONITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B3>jen(:PDT) + <B3>ja(:IMP) + <B2>(:IMT) + <B2>vši/GPP;

POSJEDOVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B5>uj(:IMT2) + <B5>ujući/GPS;

POSTAJATI = :infinitiv + <B5>oji (:Pre) + <B2>(:AO1) + <B2>(:PDR)

+ <B5>oj (im/IMT+s+m+1 + <E>/IMT+s+m+2 + i/IMT+s+m+3 + mo/IMT+p+m+1 + te/IMT+p+m+2 + e/IMT+p+m+3 +

im/IMT+s+f+1 + <E>/IMT+s+f+2 + i/IMT+s+f+3 + mo/IMT+p+f+1 + te/IMT+p+f+2 + e/IMT+p+f+3 +

im/IMT+s+n+1 + <E>/IMT+s+n+2 + i/IMT+s+n+3 + mo/IMT+p+n+1 + te/IMT+p+n+2 + e/IMT+p+n+3)

+ <B2>vši/GPP;

POZNATI = :infinitiv + :PRbirati

+ <B2> (h/AO+s+m+1 + še/AO+s+m+2 + še/AO+s+m+3 + smo/AO+p+m+1 + ste/AO+p+m+2 + hu/AO+p+m+3 +

h/AO+s+f+1 + še/AO+s+f+2 + še/AO+s+f+3 + smo/AO+p+f+1 + ste/AO+p+f+2 + hu/AO+p+f+3 +

h/AO+s+n+1 + še/AO+s+n+2 + še/AO+s+n+3 + smo/AO+p+n+1 + ste/AO+p+n+2 + hu/AO+p+n+3)

+ <B2>(:PDR) + <B1>(:PDT) + :IMTbirati + <B2>jući/GPS;

PREKINUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1) + <B4>d(:AO2) + <B2>(:PDR) + <B1>(:PDT) + <B3> (:IMT3) + <B2>vši/GPP;

PREPOZNATI = :infinitiv + :PRbirati + <B2>(:AO1) + <B2>(:PDR) + <B1>(:PDT) + :IMTbirati + <B2>vši/GPP;

PREZIRATI = :infinitiv + <B3>e(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) + <B3>(:IMT3) + <B3>ući/GPS;

PRODAVATI = :infinitiv + <B4>je(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT)

+ <B4>jem/IMT+s+m+1 + <B2>j/IMT+s+m+2 + <B4>je/IMT+s+m+3 + <B2>j(mo/IMT+p+m+1 + te/IMT+p+m+2) + <B4>ju/IMT+p+m+3 +

<B4>jem/IMT+s+f+1 + <B2>j/IMT+s+f+2 + <B4>je/IMT+s+f+3 + <B2>j(mo/IMT+p+f+1 + te/IMT+p+f+2) + <B4>ju/IMT+p+f+3 +

<B4>jem/IMT+s+n+1 + <B2>j/IMT+s+n+2 + <B4>je/IMT+s+n+3 + <B2>j(mo/IMT+p+n+1 + te/IMT+p+n+2) + <B4>ju/IMT+p+n+3

```

+ <B4>jući/GPS;

PROTEGNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1)

+ <B4>oh/AO+s+m+1 + <B5>ž(e/AO+s+m+2 + e/AO+s+m+3) + <B4>o(smo/AO+p+m+1 +
ste/AO+p+m+2 + še/AO+p+m+3) +

+ <B4>oh/AO+s+n+1 + <B5>ž(e/AO+s+n+2 + e/AO+s+n+3) + <B4>o(smo/AO+p+n+1 +
ste/AO+p+n+2 + še/AO+p+n+3) +

+ <B4>oh/AO+s+f+1 + <B5>ž(e/AO+s+f+2 + e/AO+s+f+3) + <B4>o(smo/AO+p+f+1 +
ste/AO+p+f+2 + še/AO+p+f+3)

+ <B2>(:PDR) + <B1>(:PDT) + <B3>(:IMT3) + <B2>vši/GPP;

REĆI = :infinitiv + <B2>kne(:PRu) + :PRisteći + <B2>(:AOk) + <B2>k(:PDR2) +
<B2>čen(:PDT) + <B2>kn(:IMT3) + :IMTisteći + <B2>kavši/GPP;

PRILEĆI = :infinitiv + <B2>gne(:PRu) + <B2>(:AOg) + <B2>g(:PDR2) + <B2>gn(:IMT3)

+ <B2>zi (<E>/IMT+s+m+2 + mo/IMT+p+m+1 + te/IMT+p+m+2 +

+ <E>/IMT+s+f+2 + mo/IMT+p+f+1 + te/IMT+p+f+2 +

+ <E>/IMT+s+n+2 + mo/IMT+p+n+1 + te/IMT+p+n+2)

+ <B2>gavši/GPP;

RAZLIKOVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
<B5>uj(:IMT2) + <B5>ujući/GPS;

SEZATI = :infinitiv + <B4>že(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>ž(:IMT3) +
<B4>jući/GPS;

SJATI = :infinitiv + :PRbirati + <B2>(:IMP) + <B2>(:PDR) + :IMTbirati +
<B2>jući/GPS;

SJEĆI = <E>/INF + <E>/INFk + <B4>iječe(:PRisteći) + <B4>ijecija(:IMP) +
<B2>k(:PDR2) + <B2>čen(:PDT)

+ <B4>ije (čem/IMT+s+m+1 + ci/IMT+s+m+2 + če/IMT+s+m+3 + cimo/IMT+p+m+1 +
cite/IMT+p+m+2 + ku/IMT+p+m+3 +

+ čem/IMT+s+f+1 + ci/IMT+s+f+2 + če/IMT+s+f+3 + cimo/IMT+p+f+1 +
cite/IMT+p+f+2 + ku/IMT+p+f+3 +

+ čem/IMT+s+n+1 + ci/IMT+s+n+2 + če/IMT+s+n+3 + cimo/IMT+p+n+1 +
cite/IMT+p+n+2 + ku/IMT+p+n+3)

+ <B4>ijekući/GPS;

SJESTI = :infinitiv + <B3>dne(:PRu) + <B3>do(:AO1) + <B3>(:PDR) + <B3>dn(:IMT3)
+ <B3>vši/GPP;

SLATI = :infinitiv + <BW>šalje(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +
<BW>šalj(:IMT3) + <BW>šaljući/GPS;

SLEĆI = <E>/INF + <E>/INFk + <B2>gne(:PRu) + <B2>(:AOg) + <B2>g(:PDR2) +
<B2>gnut(:PDT) + <B2>gn(:IMT3) + <B2>gavši/GPP;

SLEGNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1)

```

+ <B4>oh/AO+s+m+1 + <B5>že(<E>/AO+s+m+2 + <E>/AO+s+m+3) + <B4>o(smo/AO+p+m+1  
 + ste/AO+p+m+2 + še/AO+p+m+3) +  
  
 <B4>oh/AO+s+f+1 + <B5>že(<E>/AO+s+f+2 + <E>/AO+s+f+3) + <B4>o(smo/AO+p+f+1  
 + ste/AO+p+f+2 + še/AO+p+f+3) +  
  
 <B4>oh/AO+s+n+1 + <B5>že(<E>/AO+s+n+2 + <E>/AO+s+n+3) + <B4>o(smo/AO+p+n+1  
 + ste/AO+p+n+2 + še/AO+p+n+3)  
  
 + <B2>(:PDR) + <B1>(:PDT) + <B3>(:IMT3) + <B2>vši/GPP + <B4>avši/GPP;  
  
 SRESTI = :infinitiv + <B3>tne(:PRu) + <B3>t(:AO2) + <B3>(:PDR) + <B3>tn(:IMT3) +  
 <B3>vši/GPP;  
  
 STIDJETI = :infinitiv + <B4>i(:Pre) + <B5>đa(:IMP) + :PDRdoživjeti + <B4>i(:IMT) +  
 <B4>eći/GPS;  
  
 STIGNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1)  
  
 + <B4>oh/AO+s+m+1 + <B5>že(<E>/AO+s+m+2 + <E>/AO+s+m+3) + <B4>o(smo/AO+p+m+1  
 + ste/AO+p+m+2 + še/AO+p+m+3) +  
  
 <B4>oh/AO+s+f+1 + <B5>že(<E>/AO+s+f+2 + <E>/AO+s+f+3) + <B4>o(smo/AO+p+f+1  
 + ste/AO+p+f+2 + še/AO+p+f+3) +  
  
 <B4>oh/AO+s+n+1 + <B5>že(<E>/AO+s+n+2 + <E>/AO+s+n+3) + <B4>o(smo/AO+p+n+1  
 + ste/AO+p+n+2 + še/AO+p+n+3)  
  
 + <B2>(:PDR) + <B3>(:IMT3) + <B2>vši/GPP + <B4>avši/GPP;  
  
 STISNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1)  
  
 + <B4>koh/AO+s+m+1 + <B5>šte(<E>/AO+s+m+2 + <E>/AO+s+m+3) +  
 <B4>ko(smo/AO+p+m+1 + ste/AO+p+m+2 + še/AO+p+m+3) +  
  
 <B4>koh/AO+s+f+1 + <B5>šte(<E>/AO+s+f+2 + <E>/AO+s+f+3) +  
 <B4>ko(smo/AO+p+f+1 + ste/AO+p+f+2 + še/AO+p+f+3) +  
  
 <B4>koh/AO+s+n+1 + <B5>šte(<E>/AO+s+n+2 + <E>/AO+s+n+3) +  
 <B4>ko(smo/AO+p+n+1 + ste/AO+p+n+2 + še/AO+p+n+3)  
  
 + <B2>(:PDR) + <B1>(:PDT) + <B3>(:IMT3) + <B2>vši/GPP;  
  
 SUDITI = :infinitiv + <B2>(:Pre) + <B4>đa(:IMP) + <B2>(:PDR) + <B4>đen(:PDT) +  
 <B2>(:IMT) + <B3>eći/GPS;  
  
 SVIDJETI = :infinitiv + <B4>i(:Pre) + <B2>(:AO1) + :PDRdoživjeti + <B4>i(:IMT) +  
 <B2>vši/GPP;  
  
 SMIJATI = :infinitiv + <B3>e(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B3>(:IMT2) +  
 <B3>ući/GPS;  
  
 PATITI = :infinitiv + <B2>(:Pre) + <B4>ća(:IMP) + <B2>(:PDR) + <B4>ćen(:PDT) +  
 <B2>(:IMT) + <B3>eći/GPS;  
  
 PAZITI = :infinitiv + <B2>(:Pre) + <B4>ža(:IMP) + <B2>(:PDR) + <B4>žen(:PDT) +  
 <B2>(:IMT) + <B3>eći/GPS;  
  
 PLAKATI = :infinitiv + <B4>če(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>č(:IMT3) +  
 <B4>čući/GPS;  
  
 PLAMTJETI = :infinitiv + <B4>i(:Pre) + <B5>ča(:IMP) + :PDRdoživjeti + <B4>i(:IMT)  
 + <B4>eći/GPS;

PODIZATI = :infinitiv + <B4>že (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B4>ž (:IMT3) + <B4>žući/GPS;

POČINJATI = :infinitiv + <B3>e(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B3>(:IMT3) + <B3>ući/GPS;

POSLATI = :infinitiv + <B5>šalj (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B5>šalj(:IMT3) + <B2>vši/GPP;

PREBIRATI = :infinitiv + <B3>e(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B3>(:IMT3) + <B3>ući/GPS;

PRIJEĆI = <E>/INF + <E>/INFk + <B2>đe (:PRu) + <B2>đ(:AO2) + <B5>eš(:PDR2) +  
 <B2>đen(:PDT) + <B2>đ (:IMT3) + <B5>ešavši/GPP;

PRITISKATI = :infinitiv + <B5>šte (:PRu) + <B5>šće (:PRu) + <B2>(:IMP) + <B2>(:PDR)  
 + <B2>n(:PDT) + <B5>št (:IMT3) + <B5>šč (:IMT3) + <B5>š(tući/GPS + ćući/GPS);

PROĆI = <E>/INF + <E>/INFk + <B2>đe (:PRu) + <B2>đ(:AO2) + <B2>š(:PDR2) + <B2>đ  
 (:IMT3) + <B2>šavši/GPP;

POJESTI = :infinitiv + <B3>de (:PRu) + <B3>do (:AO1) + <B3> (:PDR) + <B3>den(:PDT)  
 + <B3>d (:IMT3) + <B3>vši/GPP;

POTRAJATI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B3> (:IMT2) +  
 <B2>vši/GPP;

PUSTITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B5>šten(:PDT) +  
 <B2>(:IMT) + <B2>vši/GPP;

RASTI = :infinitiv + <B1>e(:PRu) + ja(:IMP)  
 + <B1>(ao/PDR+s+1+m + ao/PDR+s+2+m + ao/PDR+s+3+m) + <B2> (la/PDR+s+1+f +  
 la/PDR+s+2+f + la/PDR+s+3+f + lo/PDR+s+1+n + lo/PDR+s+2+n + lo/PDR+s+3+n +  
 li/PDR+p+1+m + li/PDR+p+2+m + li/PDR+p+3+m + le/PDR+p+1+f + le/PDR+p+2+f +  
 le/PDR+p+3+f + la/PDR+p+1+n + la/PDR+p+2+n + la/PDR+p+3+n)  
 + <B1> (:IMT3) + <B1>ući/GPS;

RAZMISLITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B2>(:IMT) +  
 <B2>vši/GPP;

SJEĆI = :infinitiv + <B4>iječe(:PRisteći) + <B4>ijecija(:IMP) + <B2>k(:PDR2) +  
 <B2>čen(:PDT)  
 + <B4>ije (čem/IMT+s+m+1+ ci/IMT+s+m+2 + će/IMT+s+m+3 + cimo/IMT+p+m+1 +  
 cite/IMT+p+m+2 + ku/IMT+p+m+3 +  
 čem/IMT+s+f+1 + ci/IMT+s+f+2 + će/IMT+s+f+3 + cimo/IMT+p+f+1 +  
 cite/IMT+p+f+2 + ku/IMT+p+f+3 +  
 čem/IMT+s+n+1 + ci/IMT+s+n+2 + će/IMT+s+n+3 + cimo/IMT+p+n+1 +  
 cite/IMT+p+n+2 + ku/IMT+p+n+3)  
 + <B4>ijekući/GPS;

SJESTI = :infinitiv + <B3>dne (:PRu) + <B3>do (:AO1) + <B3> (:PDR) + <B3>dn (:IMT3)  
 + <B3>vši/GPP;

SLATI = :infinitiv + <BW>šalj (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +  
 <BW>šalj (:IMT3) + <BW>šaljući/GPS;

SLIJEGATI = :infinitiv + <B4>že (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B4>ž(:IMT3) +  
 <B4>žući/GPS;

SPATI = :infinitiv + <B3>i(:Pre) + <B2>(:IMP) + <B3>(:PDR) + B3>i(:IMT) + <B3>eći/GPS;

STAJATI = :infinitiv + <B5>oji(:Pre) + <B2>(:IMP) + <B2>(:PDR)

+ <B5>oj(im/IMT+s+m+1 + <E>/IMT+s+m+2 + i/IMT+s+m+3 + mo/IMT+p+m+1 + te/IMT+p+m+2 + e/IMT+p+m+3 +

im/IMT+s+f+1 + <E>/IMT+s+f+2 + i/IMT+s+f+3 + mo/IMT+p+f+1 + te/IMT+p+f+2 + e/IMT+p+f+3 +

im/IMT+s+n+1 + <E>/IMT+s+n+2 + i/IMT+s+n+3 + mo/IMT+p+n+1 + te/IMT+p+n+2 + e/IMT+p+n+3)

+ <B5>ojeći/GPS;

TAKNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1)

+ <B4> (oh/AO+s+m+1 + <B1>če/AO+s+m+2 + <B1>če/AO+s+m+3 + osmo/AO+p+m+1 + oste/AO+p+m+2 + oše/AO+p+m+3 +

oh/AO+s+f+1 + <B1>če/AO+s+f+2 + <B1>če/AO+s+f+3 + osmo/AO+p+f+1 + oste/AO+p+f+2 + oše/AO+p+f+3 +

oh/AO+s+n+1 + <B1>če/AO+s+n+2 + <B1>če/AO+s+n+3 + osmo/AO+p+n+1 + oste/AO+p+n+2 + oše/AO+p+n+3)

+ <B2>(:PDR) + <B1>(:PDT) + <B3>(:IMT3) + <B2>vši/GPP + <B4>avši/GPP;

TRESTI = :infinitiv + <B2>e (:PRu) + <B2>ija (:IMP) + <B2>(:PDR2) + <B2>en(:PDT) + <B2>(:IMT3) + <B2>ući/GPS;

TONUTI = :infinitiv + <B3>e(:PRu) + <B3>ja(:IMP) + <B2>(:PDR) + <B3>(:IMT3) + <B3>ući/GPS;

UGASITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B4>šen(:PDT) + <B2>(:IMT) + <B2>vši/GPP;

UGRISTI = :infinitiv + <B3>ze(:PRu) + <B3>z(:AO2) + <B3>z(:PDR2) + <B3>zen(:PDT) + <B3>z (:IMT3) + <B3>zavši/GPP;

UGOJITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B3>en(:PDT) + <B2>(:IMT) + <B2>vši/GPP;

UGOSTITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B5>šćen(:PDT) + <B2>(:IMT) + <B2>vši/GPP;

UMAKNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:AO1)

+ <B4> (oh/AO+s+m+1 + <B1>če(<E>/AO+s+m+2 + <E>/AO+s+m+3) + osmo/AO+p+m+1 + oste/AO+p+m+2 + oše/AO+p+m+3) +

<B4> (oh/AO+s+f+1 + <B1>če(<E>/AO+s+f+2 + <E>/AO+s+f+3) + osmo/AO+p+f+1 + oste/AO+p+f+2 + oše/AO+p+f+3) +

<B4> (oh/AO+s+n+1 + <B1>če(<E>/AO+s+n+2 + <E>/AO+s+n+3) + osmo/AO+p+n+1 + oste/AO+p+n+2 + oše/AO+p+n+3)

+ <B3>(:IMT3) + <B2>vši/GPP + <B4>avši/GPP;

UMJETI = :infinitiv + <B4>ije (:PRu) + <B2>(:AO1) + :PDRdoživjeti + <B4>ij(:IMT2) + <B2>vši/GPP;

```

UMRIJETI = :infinitiv + <B5>e (:PRu) + <B2>(:AO1) + <B5>(:PDR) + <B5>(:IMT3) +
<B2>vši/GPP;

USISATI = :infinitiv + <B4>še(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +
<B4>š(:IMT3) + <B2>vši/GPP;

UZETI = :infinitiv + <B3>me(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B1>(:PDT) +
<B3>m(:IMT3) + <B2>vši/GPP;

VESTI = :infinitiv + <B3>ze (:PRu) + <B3>zija (:IMP) + <B3>z(:PDR2) + <B3>zen(:PDT)
+ <B3>z(:IMT3) + <B3>zući/GPS;

VIDJETI = :infinitiv + <B4>i (:Pre) + <B2>(:AO1) + :PDRdoživjeti + <B5>đen(:PDT) +
<B4>i(:IMT) + <B2>vši/GPP;

VRTJETI = :infinitiv + <B4>i(:Pre) + <B5>ća(:IMP) + :PDRdoživjeti + <B5>ćen(:PDT) +
<B4>i(:IMT) + <B4>eći/GPS;

ZABITI = :infinitiv + <B2>je(:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>jen(:PDT) +
<B1>(:PDT) + <B2>j(:IMT2) + <B2>vši/GPP;

ZADRŽATI = :infinitiv + <B3>i(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +
B3>i(:IMT) + <B2>vši/GPP;

ZAHVALJIVATI = :infinitiv + <B5>uje(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B5>uj(:IMT2)
+ <B5>ujući/GPS;

ZAMISLITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B5>šljen(:PDT) +
<B2>(:IMT) + <B2>vši/GPP;

ZANIJETI = :infinitiv + <B5>ese(:PRu) + <B5>es(:AO2) + <B2>(:AO1)
+ <B4>(o/PDR+s+1+m + o/PDR+s+2+m + o/PDR+s+3+m) + <B2>l (a/PDR+s+1+f +
a/PDR+s+2+f + a/PDR+s+3+f + o/PDR+s+1+n + o/PDR+s+2+n + o/PDR+s+3+n + i/PDR+p+1+m +
i/PDR+p+2+m + i/PDR+p+3+m + e/PDR+p+1+f + e/PDR+p+2+f + e/PDR+p+3+f + a/PDR+p+1+n +
a/PDR+p+2+n + a/PDR+p+3+n)
+ <B1>(:PDT) + <B5>esen(:PDT) + <B5>es(:IMT3) + <B2>vši/GPP;

ZAPAZITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B4>žen(:PDT) +
<B2>(:IMT) + <B2>vši/GPP;

ZAPEĆI = <E>/INF + <E>/INFk + :PRisteći + <B2>(:AOk) + <B2>k(:PDR2) + <B2>čen(:PDT)
+ :IMTisteći + <B2>kavši/GPP;

ZAPLJESKATI = :infinitiv + <B5>šće (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B5>šč(:IMT3)
+ <B2>vši/GPP;

ZAPOČETI = :infinitiv + <B3>ne (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B1>(:PDT) +
<B3>n(:IMT3) + <B2>vši/GPP;

ZASJATI = :infinitiv + :PRbirati + <B2>(:AO1) + <B2>(:PDR)
+ <B2> (m/IMT+s+m+1 + j/IMT+s+m+2 + <E>/IMT+s+m+3 + mo/IMT+p+m+1 +
te/IMT+p+m+2 + ju/IMT+p+m+3 +
m/IMT+s+f+1 + j/IMT+s+f+2 + <E>/IMT+s+f+3 + mo/IMT+p+f+1 +
te/IMT+p+f+2 + ju/IMT+p+f+3 +
m/IMT+s+n+1 + j/IMT+s+n+2 + <E>/IMT+s+n+3 + mo/IMT+p+n+1 +
te/IMT+p+n+2 + ju/IMT+p+n+3)
+ <B2>vši/GPP;

```

ZASPATI = :infinitiv + <B3>i (:PRu) + <B3>e/PR+p+m+3 + <B3>e/PR+p+f+3 +  
 <B3>e/PR+p+n+3 + <B2>(:AO1) + <B2>(:PDR) + <B3>i(:IMT) + <B3>u/IMT+p+m+3 +  
 <B3>u/IMT+p+f+3 + <B3>u/IMT+p+n+3 + <B2>vši/GPP;

ZASTATI = :infinitiv + <B2>ne (:PRu) + <B2>de(:PRu) + <B2>d(:AO2) + <B2>(:PDR) +  
 <B2>n (:IMT3)

+ <B2>d (em/IMT+s+m+1 + e/IMT+s+m+3 + u/IMT+p+m+3 +  
 em/IMT+s+f+1 + e/IMT+s+f+3 + u/IMT+p+f+3 +  
 em/IMT+s+n+1 + e/IMT+s+n+3 + u/IMT+p+n+3)

+ <B2>vši/GPP;

ZASUTI = :infinitiv + <B3>pe (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B1>(:PDT) + <B3>p  
 (:IMT3) + <B2>vši/GPP;

ZATRESTI = :infinitiv + <B2>e (:PRu) + <B2>(:AO2) + <B2>(:PDR2) + <B2>en(:PDT) +  
 <B2>(:IMT3) + <B2>avši/GPP;

ZATUTNJATI = :infinitiv + + <B3>i (:Pre) + <B2>(:AO1) + <B2>(:PDR)

+ <B2> (m/IMT+s+m+1 + j/IMT+s+m+2 + <B1>i(<E>/IMT+s+m+3 + mo/IMT+p+m+1 +  
 te/IMT+p+m+2) + ju/IMT+p+m+3 +

m/IMT+s+f+1 + j/IMT+s+f+2 + <B1>i(<E>/IMT+s+f+3 + mo/IMT+p+f+1 +  
 te/IMT+p+f+2) + ju/IMT+p+f+3 +

m/IMT+s+n+1 + j/IMT+s+n+2 + <B1>i(<E>/IMT+s+n+3 + mo/IMT+p+n+1 +  
 te/IMT+p+n+2) + ju/IMT+p+n+3)

+ <B2>vši/GPP;

ZAUSTITI = :infinitiv + <B2>(:Pre) + <B2>(:AO1) + <B2>(:PDR) + <B2>(:IMT) +  
 <B2>vši/GPP;

ZAZVATI = :infinitiv + <B4>ove (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B4>ov(:IMT3) + <B2>vši/GPP;

ZAŽELJETI = :infinitiv + <B4>i (:Pre) + <B2>(:AO1) + :PDRdoživjeti

+ <B4>i(m/IMT+s+m+1 + <E>/IMT+s+m+2 + <E>/IMT+s+m+3 + imo/IMT+p+m+1 +  
 ite/IMT+p+m+2 + <B1>e/IMT+p+m+3 +

m/IMT+s+f+1 + <E>/IMT+s+f+2 + <E>/IMT+s+f+3 + imo/IMT+p+f+1 +  
 ite/IMT+p+f+2 + <B1>e/IMT+p+f+3 +

m/IMT+s+n+1 + <E>/IMT+s+n+2 + <E>/IMT+s+n+3 + imo/IMT+p+n+1 +  
 ite/IMT+p+n+2 + <B1>e/IMT+p+n+3)

+ <B2>vši/GPP;

ZNATI = :infinitiv + :PRbirati + <B2>de(:PRu) + <B2>(:IMP) + <B2>diја(:IMP) +  
 <B2>(:PDR) + <B2>n(:PDT) + :IMTbirati + :IMTdatil + <B2>jući/GPS;

ZRETI = :infinitiv + <B3>ije(:PRu) + <B3>ija(:IMP)

+ <B3>e (<B1>(io/PDR+s+1+m + io/PDR+s+2+m + io/PDR+s+3+m) + la/PDR+s+1+f +  
 la/PDR+s+2+f + la/PDR+s+3+f + lo/PDR+s+1+n + lo/PDR+s+2+n + lo/PDR+s+3+n +  
 li/PDR+p+1+m + li/PDR+p+2+m + li/PDR+p+3+m + le/PDR+p+1+f + le/PDR+p+2+f +  
 le/PDR+p+3+f + la/PDR+p+1+n + la/PDR+p+2+n + la/PDR+p+3+n)



+ <B3>ij (em/IMT+s+m+1 + <B1>/IMT+s+m+2 + e/IMT+s+m+3 + <B1>mo/IMT+p+m+1 +  
 <B1>te/IMT+p+m+2 + u/IMT+p+m+3 +  
 em/IMT+s+n+1 + <B1>/IMT+s+n+2 + e/IMT+s+n+3 + <B1>mo/IMT+p+n+1 +  
 <B1>te/IMT+p+n+2 + u/IMT+p+n+3 +  
 em/IMT+s+f+1 + <B1>/IMT+s+f+2 + e/IMT+s+f+3 + <B1>mo/IMT+p+f+1 +  
 <B1>te/IMT+p+f+2 + u/IMT+p+f+3)  
 + <B3>ijući/GPS;

ZVATI = :infinitiv + <B4>ove (:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B4>ov(:IMT3) + <B4>ovući/GPS;

ŽAGORITI = :infinitiv + <B2>(:Pre) + <B3>a(:IMP) + <B2>(:PDR) + <B2>(:IMT) +  
 <B3>eći/GPS;

VOLJETI = :infinitiv + <B4>i(:Pre) + <B3>a(:IMP) + :PDRdoživjeti + <B2>n(:PDT) +  
 <B4>i(:IMT) + <B4>eći/GPS;

ZAPLAKATI = :infinitiv + <B4>če (:PRu) + <B2>(:AO1) + <B2>(:PDR) + <B4>č(:IMT3) +  
 <B2>vši/GPP;

ŽIVJETI = :infinitiv + <B4>i(:Pre) + <B4>lja(:IMP) + :PDRdoživjeti + <B4>i(:IMT) +  
 <B4>eći/GPS;

ŽVAKATI = :infinitiv + <B4>če(:PRu) + <B2>(:IMP) + <B2>(:PDR) + <B2>n(:PDT) +  
 <B4>č(:IMT3) + <B4>čući/GPS;

GUBITI = :infinitiv + <B2>(:Pre) + <B3>lja(:IMP) + <B2>(:PDR) + <B3>ljen(:PDT) +  
 <B2>(:IMT) + <B3>eći/GPS;

ISTEĆI = :infinitiv + :PRisteći + <B2>k(:PDR2) + <B2>(:AOk) + :IMTisteći +  
 <B2>kavši/GPP;

IZDATI = :infinitiv + :PRbirati + <B2>(:PDR) + <B2>d(:AO2) + <B2>n(:PDT) +  
 :IMTbirati + <B2>vši/GPP;

IZLEĆI = :infinitiv + :PRizleći + <B2>g(:PDR2) + <B2>(:AOg) + <B2>žen(:PDT) +  
 :IMTizleći + <B2>gavši/GPP;

KRITI = :infinitiv + <B2>je(:PRu) + <B2>ja(:IMP) + <B2>(:PDR) + <B2>j(:IMT2) +  
 <B2>ven(:PDT) + <B2>jući/GPS;

LEŽATI = :infinitiv + <B3>i(:Pre) + <B2>(:IMP) + <B2>(:PDR) + <B3>i(:IMT) +  
 <B3>eći/GPS;

MLJETI = :infinitiv + ((<B2>(:PRu))<LW><R1>e) + ((<B3>a(:IMP))<LW><R1>e) +  
 <B4>i(o/PDR+s+1+m + o/PDR+s+2+m + o/PDR+s+3+m + la/PDR+s+1+f + la/PDR+s+2+f  
 + la/PDR+s+3+f + lo/PDR+s+1+n + lo/PDR+s+2+n + lo/PDR+s+3+n) + <B2>(li/PDR+p+1+m +  
 li/PDR+p+2+m + li/PDR+p+3+m + le/PDR+p+1+f + le/PDR+p+2+f + le/PDR+p+3+f +  
 la/PDR+p+1+n + la/PDR+p+2+n + la/PDR+p+3+n) +  
 ((<B3>(:IMT3))<LW><R1>e) + <B2>ven(:PDT) + ((<B3>ući/GPS)<LW><R1>e);

NAPOMINJATI = :infinitiv + <B3>e(:PRu) + <B2>(:PDR) + <B2>(:IMP) + <B2>n(:PDT) +  
 <B3>(:IMT3) + <B3>ući/GPS;

NAPUHATI = :infinitiv + <B4>še(:PRu) + <B2>(:PDR) + <B2>(:AO1) + <B2>n(:PDT) +  
 <B4>š(:IMT3) + <B2>vši/GPP;

NASUKATI = :infinitiv + <B4>če(:PRu) + <B2>(:PDR) + <B2>(:AO1) + <B2>n(:PDT) +  
 <B4>č(:IMT3) + <B2>vši/GPP;

```

ODLITI = :infinitiv + <B2>je(:PRu) + <B2>(:PDR) + <B2>(:AO1) + <B2>ven(:PDT) +
<B1>(:PDT) + <B2>j(:IMT2) + <B2>vši/GPP;

OSVRNUTI = :infinitiv + <B3>e(:PRu) + <B2>(:PDR) + <B2>(:AO1) + <B4>t(:AO2) +
<B3>(:IMT3) + <B2>vši/GPP;

OTPRIJETI = :infinitiv + <B5>e(:PRu) + <B5>(:PDR) + AObljesnuti + <B5>t(:PDT) +
<B2>vši/GPP + <B5>(:IMT3);

POBJEĆI = :infinitiv + <B2>gne(:PRu) + <B2>g(:PDR2) + <B2>(:AOg) + <B2>gn(:IMT3) +
<B2>gavši/GPP;

PRODATI = :infinitiv + :PRbirati + <B2>(:PDR) + <B2>(:AO1) + <B2>n(:PDT) +
:IMTbirati + <B2>vši/GPP;

POSTITI = :infinitiv + <B2>(:Pre) + <B2>(:PDR) + <B5>šća(:IMP) + <B2>(:IMT) +
<B3>eći/GPS;

PREDSTOJATI = :infinitiv + <B3>i(:Pre) <B2>(:PDR) + <B2>(:IMP) +

                <B3>(im/IMT+s+m+1 + <E>/IMT+s+m+2 + i/IMT+s+m+3 + mo/IMT+p+m+1 +
te/IMT+p+m+2 + e/IMT+p+m+3 +

                im/IMT+s+f+1 + <E>/IMT+s+f+2 + i/IMT+s+f+3 + mo/IMT+p+f+1 +
te/IMT+p+f+2 + e/IMT+p+f+3 +

                im/IMT+s+n+1 + <E>/IMT+s+n+2 + i/IMT+s+n+3 + mo/IMT+p+n+1 +
te/IMT+p+n+2 + e/IMT+p+n+3)

                + <B3>eći/GPS;

RASPASTI = :infinitiv + <B3>dne(:PRu) + <B3>(:PDR) + <B3>d(AO2) + <B3>dnut(:PDT) +
<B3>dn(:IMT3) + <B3>vši/GPP;

SAŽETI = :infinitiv + <B3>me(:PRu) + <B2>(:PDR) + <B2>(:AO1) + <B1>(:PDT) +
<B3>m(:IMT3) + <B2>vši/GPP;

UGINUTI = :infinitiv + <B3>e(:PRu) + <B2>(:PDR) + <B2>(:AO1) + <B3>(:IMT3) +
<B2>vši/GPP;

VELIM = <B1>(:Pre);

VISJETI = :infinitiv + <B4>i(:Pre) + :PDRdoživjeti + <B5>š(:IMP) + <B4>i(:IMT) +
<B4>eći/GPS;

VUĆI = :infinitiv + :PRisteći + <B2>k(:PDR2) + <B2>cija(:IMP) + <B2>čen(:PDT) +
:IMTisteći + <B2>kući/GPS;

ZASIPATI = :infinitiv + <B3>lje(:PRu) + <B2>(:PDR) + <B2>(:IMP) + <B2>n(:PDT) +
<B3>lj(:IMT3) + <B3>ljući/GPS;

ZATEĆI = :infinitiv + <B2>kne(:PRu) + <B2>k(:PDR2) + <B2>(:AOk) + <B2>kn(:IMT3) +
<B2>kavši/GPP;

```

## Prilog 6 – Opis paradigmi za brojeve i zamjenice (hrCroNumPro.nof)

**JEDAN** = <E>/Nom+m+s + <E>/Acc+m+s + <E>/Voc+m+s + <B2>no (g/G+m+s + ga/G+m+s + m/D+m+s + mu/D+m+s + me/D+m+s + m/L+m+s + mu/L+m+s + me/L+m+s + <B1>im/I+m+s)

+ <B2>no (<E>/Nom+n+s + <E>/Acc+n+s + <E>/Voc+n+s) + <B2>no (g/G+n+s + ga/G+n+s + m/D+n+s + mu/D+n+s + me/D+n+s + m/L+n+s + mu/L+n+s + me/L+n+s + <B1>im/I+n+s)

+ <B2>na (<E>/Nom+f+s + <E>/Voc+f+s) + <B2>n (e/G+f+s + oj/D+f+s + u/Acc+f+s + oj/L+f+s + om/I+f+s);

**DVA** = <E>/Nom+m+n+p + <E>/Acc+m+n+p + <E>/Voc+m+n+p + ju/G+m+n+p + ma/D+m+n+p + ma/L+m+n+p + ma/I+m+n+p

+ <E>/Nom+n+p + <E>/Acc+n+p + <E>/Voc+n+p + ju/G+n+p + ma/D+n+p + ma/L+n+p + ma/I+n+p

+ <B1>ije(<E>/Nom+f+p + <E>/Acc+f+p + <E>/Voc+f+p) + <B1>iju/G+f+p + <B1>jema(<E>/D+f+p + <E>/L+f+p + <E>/I+f+p);

**TRI** = <E>/Nom+m+p + <E>/Acc+m+p + <E>/Voc+m+p + ju/G+m+p + ma/D+m+p + ma/L+m+p + ma/I+m+p

+ <E>/Nom+n+p + <E>/Acc+n+p + <E>/Voc+n+p + ju/G+n+p + ma/D+n+p + ma/L+n+p + ma/I+n+p

+ <E>/Nom+f+p + <E>/Acc+f+p + <E>/Voc+f+p + ju/G+f+p + ma/D+f+p + ma/L+f+p + ma/I+f+p;

**PRVI** = <E>/Nom+m+s + <B1>(og/G+m+s + oga/G+m+s + om/D+m+s + ome/D+m+s + omu/D+m+s + og/Acc+m+s + oga/Acc+m+s + i/Voc+m+s + om/L+m+s + ome/L+m+s + omu/L+m+s + im/I+m+s) + <E>/Acc+m+s +

<E>/Nom+m+p + h/G+m+p + m/D+m+p + ma/D+m+p + <B1>e/Acc+m+p + <E>/Voc+m+p + m/L+m+p + ma/L+m+p + m/I+m+p + ma/I+m+p

+ <B1>(a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + a/Voc+f+s + oj/L+f+s + om/I+f+s +

e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + e/Voc+f+p + im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

+ <B1>(o/Nom+n+s + og/G+n+s + oga/G+n+s + om/D+n+s + ome/D+n+s + omu/D+n+s + o/Acc+n+s + o/Voc+n+s + om/L+n+s + ome/L+n+s + omu/L+n+s + im/I+n+s +

a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + a/Voc+n+p + im/L+n+p + ima/L+n+p + im/I+n+p + ima/I+n+p);

**TREĆI** = <E>/Nom+m+s + <B1>(eg/G+m+s + ega/G+m+s + em/D+m+s + emu/D+m+s + eg/Acc+m+s + ega/Acc+m+s + i/Voc+m+s + em/L+m+s + emu/L+m+s + im/I+m+s) + <E>/Acc+m+s +

<E>/Nom+m+p + h/G+m+p + m/D+m+p + ma/D+m+p + <B1>e/Acc+m+p + <E>/Voc+m+p + m/L+m+p + ma/L+m+p + m/I+m+p + ma/I+m+p

```

    + <B1>(a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + a/Voc+f+s + oj/L+f+s +
om/I+f+s +

    e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + e/Voc+f+p +
im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

    + <B1>(e/Nom+n+s + eg/G+n+s + ega/G+n+s + em/D+n+s + emu/D+n+s + e/Acc+n+s +
e/Voc+n+s + em/L+n+s + emu/L+n+s + im/I+n+s +

    a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + a/Voc+n+p +
im/L+n+p + ima/L+n+p + im/I+n+p + ima/I+n+p);

DVOJE = <B1>(i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + i/Voc+m+p +
im/L+m+p + ima/L+m+p + im/I+m+p + ima/I+m+p)

    + <E>/Nom+f+p + <B1>(ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p +
e/Voc+f+p + im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

    + <B1>(a/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + a/Acc+f+p + a/Voc+f+p
+ im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p);

#Zamjenice

FSmoj = a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + a/Voc+f+s + oj/L+f+s +
om/I+f+s;

FPmoj = e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + e/Voc+f+p +
im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p;

NPmoj = a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + a/Voc+n+p +
im/L+n+p + ima/L+n+p + im/I+n+p + ima/L+n+p;

MPmoj = i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + i/Voc+m+p +
im/L+m+p + ima/L+m+p + im/I+m+p + ima/L+m+p;

#osobne

JA = <E>/Nom+m+s + <B2>me (ne/G+m+s + <E>/G+m+s + ni/D+m+s + <B1>i/D+m+s +
ne/Acc+m+s + <E>/Acc+m+s + ni/L+m+s + <B1>(nom/I+m+s + nome/I+m+s))

    + <E>/Nom+f+s + <B2>me (ne/G+f+s + <E>/G+f+s + ni/D+f+s + <B1>i/D+f+s +
ne/Acc+f+s + <E>/Acc+f+s + ni/L+f+s + <B1>(nom/I+f+s + nome/I+f+s))

    + <E>/Nom+n+s + <B2>me (ne/G+n+s + <E>/G+n+s + ni/D+n+s + <B1>i/D+n+s +
ne/Acc+n+s + <E>/Acc+n+s + ni/L+n+s + <B1>(nom/I+n+s + nome/I+n+s));

TI = <E>/Nom+m+s + <B1>e (be/G+m+s + <E>/G+m+s + bi/D+m+s + <B1>i/D+m+s +
be/Acc+m+s + <E>/Acc+m+s) + <E>/Voc+m+s + <B1>ebi/L+m+s + <B1>obom/I+m+s

    + <E>/Nom+f+s + <B1>e (be/G+f+s + <E>/G+f+s + bi/D+f+s + <B1>i/D+f+s +
be/Acc+f+s + <E>/Acc+f+s) + <E>/Voc+f+s + <B1>ebi/L+f+s + <B1>obom/I+f+s

    + <E>/Nom+n+s + <B1>e (be/G+n+s + <E>/G+n+s + bi/D+n+s + <B1>i/D+n+s +
be/Acc+n+s + <E>/Acc+n+s) + <E>/Voc+n+s + <B1>ebi/L+n+s + <B1>obom/I+n+s;

ON = <E>/Nom+m+s + <BW>nje (ga/G+m+s + mu/D+m+s + ga/Acc+m+s + <B1>/Acc+m+s +
mu/L+m+s + m/L+m+s + <B1>i(m/I+m+s + me/I+m+s)) + ga/G+m+s + mu/D+m+s + ga/Acc+m+s

    + i/Nom+m+p + <BW>nji(h/G+m+p + ma/D+m+p + h/Acc+m+p + ma/L+m+p + ma/I+m+p) +
<BW>(ih/G+m+p + im/D+m+p + ih/Acc+m+p);

```

**ONA** = <E>/Nom+f+s + <BW>nj (e/G+f+s + oj/D+f+s + u/Acc+f+s + oj/L+f+s + om/I+f+s + ome/I+f+s) + <BW>j(e/G+f+s + oj/D+f+s + u/Acc+f+s + e/Acc+f+s)

+ <B1>e/Nom+f+p + <BW>nji(h/G+f+p + ma/D+f+p + h/Acc+f+p + ma/L+f+p + ma/I+f+p) + <BW>(ih/G+f+p + im/D+f+p + ih/Acc+f+p);

**ONO** = <E>/Nom+n+s + <BW>nj (ega/G+n+s + emu/D+n+s + ega/Acc+n+s + em/L+n+s + emu/L+n+s + im/I+n+s + ime/I+n+s) + <BW>(ga/G+n+s + mu/D+n+s + ga/Acc+n+s) + <BW>nj/Acc+n+s

+ <B1>a/Nom+n+p + <BW>nji(h/G+n+p + ma/D+n+p + h/Acc+n+p + ma/L+n+p + ma/I+n+p) + <BW>(ih/G+n+p + im/D+n+p + ih/Acc+n+p);

**MI** = <E>/Nom+m+p + <B2>na(s/G+m+p + ma/D+m+p + m/D+m+p + s/Acc+m+p + ma/L+m+p + ma/I+m+p)

+ <E>/Nom+f+p + <B2>na(s/G+f+p + ma/D+f+p + m/D+f+p + s/Acc+f+p + ma/L+f+p + ma/I+f+p)

+ <E>/Nom+n+p + <B2>na(s/G+n+p + ma/D+n+p + m/D+n+p + s/Acc+n+p + ma/L+n+p + ma/I+n+p);

**VI** = <E>/Nom+m+p + <E>/Voc+m+p + <B1>a(s/G+m+p + ma/D+m+p + m/D+m+p + s/Acc+m+p + ma/L+m+p + ma/I+m+p)

+ <E>/Nom+f+p + <E>/Voc+f+p + <B1>a(s/G+f+p + ma/D+f+p + m/D+f+p + s/Acc+f+p + ma/L+f+p + ma/I+f+p)

+ <E>/Nom+n+p + <E>/Voc+n+p + <B1>a(s/G+n+p + ma/D+n+p + m/D+n+p + s/Acc+n+p + ma/L+n+p + ma/I+n+p);

#povratne

**SE** = be/G+m+s + <E>/Gk+m+s + bi/D+m+s + <B1>i/D+m+s + be/Acc+m+s + <E>/Acck+m+s + bi/L+m+s + <B1>obom/I+m+s

+ be/G+f+s + <E>/Gk+s + bi/D+f+s + <B1>i/D+f+s + be/Acc+f+s + <E>/Acck+f+s + bi/L+f+s + <B1>obom/I+f+s

+ be/G+n+s + <E>/Gk+n+s + bi/D+n+s + <B1>i/D+n+s + be/Acc+n+s + <E>/Acck+n+s + bi/L+n+s + <B1>obom/I+n+s;

**SEBE** = <E>/G+m+s + <B2>/Gk+m+s + <B1>i/D+m+s + <B3>i/D+m+s + <E>/Acc+m+s + <B2>/Acck+m+s + <B1>i/L+m+s + <B3>obom/I+m+s

+ <E>/G+f+s + <B2>/Gk+s + <B1>i/D+f+s + <B3>i/D+f+s + <E>/Acc+f+s + <B2>/Acck+f+s + <B1>i/L+f+s + <B3>obom/I+f+s

+ <E>/G+n+s + <B2>/Gk+n+s + <B1>i/D+n+s + <B3>i/D+n+s + <E>/Acc+n+s + <B2>/Acck+n+s + <B1>i/L+n+s + <B3>obom/I+n+s;

#posvojne

**MOJ** = <E>/Nom+m+s + eg/G+m+s + ega/G+m+s + <B1>(ga/G+m+s + g/G+m+s) + em/D+m+s + emu/D+m+s + <B1>(m/D+m+s + me/D+m+s) + eg/Acc+m+s + ega/Acc+m+s + <B1>(g/Acc+m+s + ga/Acc+m+s) + <E>/Acc+m+s + <E>/Voc+m+s + em/L+m+s + emu/L+m+s + <B1>(m/L+m+s + me/L+m+s) + im/I+m+s

+ :MPmoj + :FSmoj + :FPmoj

+ e/Nom+n+s + eg/G+n+s + ega/G+n+s + <B1>(ga/G+n+s + g/G+n+s) + em/D+n+s + emu/D+n+s + <B1>(m/D+n+s + me/D+n+s) + e/Acc+n+s + e/Voc+n+s + em/L+n+s + emu/L+n+s + <B1>(m/L+n+s + me/L+n+s) + im/I+n+s

+ :NPmoj;

**NJEGOV** = <E>/Nom+m+s + a/G+m+s + og/G+m+s + oga/G+m+s + u/D+m+s + om/D+m+s + ome/D+m+s + omu/D+m+s + a/Acc+m+s + og/Acc+m+s + oga/Acc+m+s + <E>/Acc+m+s + <E>/Voc+m+s + u/L+m+s + om/L+m+s + ome/L+m+s + omu/L+m+s + im/I+m+s

+ :MPmoj + :FSmoj + :FPmoj

+ o/Nom+n+s + a/G+n+s + og/G+n+s + oga/G+n+s + u/D+n+s + om/D+n+s + ome/D+n+s + omu/D+n+s + o/Acc+n+s + o/Voc+n+s + u/L+n+s + om/L+n+s + ome/L+n+s + omu/L+n+s + im/I+n+s

+ :NPmoj;

**NAŠ** = <E>/Nom+m+s + eg/G+m+s + ega/G+m+s + em/D+m+s + emu/D+m+s + eg/Acc+m+s + ega/Acc+m+s + <E>/Acc+m+s + <E>/Voc+m+s + em/L+m+s + emu/L+m+s + im/I+m+s

+ :MPmoj + :FSmoj + :FPmoj

+ e/Nom+n+s + eg/G+n+s + ega/G+n+s + em/D+n+s + emu/D+n+s + e/Acc+n+s + e/Voc+n+s + em/L+n+s + emu/L+n+s + im/I+n+s

+ :NPmoj;

#pokazne

**OVAJ** = <E>/Nom+m+s + <B2>(og/G+m+s + oga/G+m+s + om/D+m+s + omu/D+m+s + ome/D+m+s + og/Acc+m+s + oga/Acc+m+s) + <E>/Acc+m+s + <B2>(om/L+m+s + omu/L+m+s + ome/L+m+s + im/I+m+s + ime/I+m+s)

+ <B2>(i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + im/L+m+p + ima/L+m+p + im/I+m+p + ima/L+m+p)

+ <B1>/Nom+f+s + <B2>(e/G+f+s + oj/D+f+s + u/Acc+f+s + oj/L+f+s + om/I+f+s

+ e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

+ <B2>(o/Nom+n+s + og/G+n+s + oga/G+n+s + om/D+n+s + omu/D+n+s + ome/D+n+s + o/Acc+n+s + om/L+n+s + omu/L+n+s + ome/L+n+s + im/I+n+s)

+ <B1>/Nom+n+p + <B2>(ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + im/L+n+p + ima/L+n+p + im/I+n+p + ima/L+n+p);

**OVAKAV** = <E>/Nom+m+s + <B2>v(a/G+m+s + og/G+m+s + oga/G+m+s + u/D+m+s + om/D+m+s + omu/D+m+s + ome/D+m+s + a/Acc+m+s + og/Acc+m+s + oga/Acc+m+s) + <E>/Acc+m+s + <B2>v(u/L+m+s + om/L+m+s + omu/L+m+s + ome/L+m+s + im/I+m+s)

+ <B2>v(i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + im/L+m+p + ima/L+m+p + im/I+m+p + ima/L+m+p)

+ <B2>v(a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + oj/L+f+s + om/I+f+s

+ e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

+ <B2>v(o/Nom+n+s + a/G+n+s + og/G+n+s + oga/G+n+s + u/D+n+s + om/D+n+s +  
omu/D+n+s + ome/D+n+s + o/Acc+n+s + u/L+n+s + om/L+n+s + omu/L+n+s + ome/L+n+s +  
im/I+n+s

+ a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + im/L+n+p +  
ima/L+n+p + im/I+n+p + ima/L+n+p);

**OVOLIK** = <E>/Nom+m+s + a/G+m+s + u/D+m+s + a/Acc+m+s + <E>/Acc+m+s + i/Voc+m+s +  
u/L+m+s + im/I+m+s

+ :MPmoj + :FSmoj + :FPmoj

+ o/Nom+n+s + a/G+n+s + u/D+n+s + o/Acc+n+s + o/Voc+n+s + u/L+n+s + im/I+n+s

+ :NPmoj;

#upitne i odnosne

**TKO** = <E>/Nom+m+s + <B3>k(oga/G+m+s + og/G+m+s + omu/D+m+s + ome/D+m+s + om/D+m+s +  
oga/Acc+m+s + og/Acc+m+s + ome/L+m+s + om/L+m+s +im/I+m+s + ime/I+m+s)

+ <E>/Nom+f+s + <B3>k(oga/G+f+s + og/G+f+s + omu/D+f+s + ome/D+f+s + om/D+f+s +  
oga/Acc+f+s + og/Acc+f+s + ome/L+f+s + om/L+f+s +im/I+f+s + ime/I+f+s)

+ <E>/Nom+n+s + <B3>k(oga/G+n+s + og/G+n+s + omu/D+n+s + ome/D+n+s + om/D+n+s +  
oga/Acc+n+s + og/Acc+n+s + ome/L+n+s + om/L+n+s +im/I+n+s + ime/I+n+s);

**ŠTO** = <E>/Nom+m+s + <B3>č(ega/G+m+s + eg/G+m+s + emu/D+m+s) + <E>/Acc+m+s +  
<B3>č(em/L+m+s + emu/L+m+s + im/I+m+s + ime/I+m+s)

+ <E>/Nom+f+s + <B3>č(ega/G+f+s + eg/G+f+s + emu/D+f+s) + <E>/Acc+f+s +  
<B3>č(em/L+f+s + emu/L+f+s + im/I+f+s + ime/I+f+s)

+ <E>/Nom+n+s + <B3>č(ega/G+n+s + eg/G+n+s + emu/D+n+s) + <E>/Acc+n+s +  
<B3>č(em/L+n+s + emu/L+n+s + im/I+n+s + ime/I+n+s);

**KOJI** = <E>/Nom+m+s + <B1>(ega/G+m+s + eg/G+m+s + em/D+m+s + emu/D+m+s + ega/Acc+m+s  
+ eg/Acc+m+s) + <E>/Acc+m+s + <B1>(em/L+m+s + emu/L+m+s + im/I+m+s)

+ <E>/Nom+m+p + h/G+m+p + m/D+m+p + ma/D+m+p + <B1>e/Acc+m+p + m/L+m+p +  
ma/L+m+p + m/I+m+p + ma/L+m+p

+ <B1>(a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + oj/L+f+s + om/I+f+s

+ e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + im/L+f+p +  
ima/L+f+p + im/I+f+p + ima/I+f+p)

+ <B1>(e/Nom+n+s + ega/G+n+s + eg/G+n+s + emu/D+n+s + em/D+n+s + e/Acc+n+s +  
emu/L+n+s + em/L+n+s + im/I+n+s)

+ <B1>(a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + im/L+n+p +  
ima/L+n+p + im/I+n+p + ima/L+n+p);

**KAKAV** = <E>/Nom+m+s + <B2>v(a/G+m+s + og/G+m+s + oga/G+m+s + u/D+m+s + ome/D+m+s +  
omu/D+m+s + a/Acc+m+s + og/Acc+m+s + oga/Acc+m+s) + <E>/Acc+m+s + <B2>v(u/L+m+s +  
om/L+m+s + omu/L+m+s + ome/L+m+s + im/I+m+s)

+ <B2>v(i/Nom+m+p + ih/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + im/L+m+p +  
ima/L+m+p + im/I+m+p + ima/L+m+p)

+ <B2>v(a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + oj/L+f+s + om/I+f+s

+ e/Nom+f+p + ih/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

+ <B2>v(o/Nom+n+s + a/G+n+s + og/G+n+s + oga/G+n+s + u/D+n+s + om/D+n+s + omu/D+n+s + ome/D+n+s + o/Acc+n+s + u/L+n+s + om/L+n+s + omu/L+n+s + ome/L+n+s + im/I+n+s)

+ <B2>v(a/Nom+n+p + ih/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + im/L+n+p + ima/L+n+p + im/I+n+p + ima/L+n+p);

**SAV** = <E>/Nom+m+s + <B2>v(ega/G+m+s + emu/D+m+s + ega/Acc+m+s) + <E>/Acc+m+s + <E>/Voc+m+s + <B2>v(em/L+m+s + emu/L+m+s + im/I+m+s + ime/I+m+s)

+<B2>v(i/Nom+m+p + ih/G+m+p + iju/G+m+p + im/D+m+p + ima/D+m+p + e/Acc+m+p + i/Voc+m+p + im/L+m+p + ima/L+m+p + im/I+m+p + ima/L+m+p)

+<B2>v(a/Nom+f+s + e/G+f+s + oj/D+f+s + u/Acc+f+s + a/Voc+f+s + oj/L+f+s + om/I+f+s)

+<B2>v(e/Nom+f+p + ih/G+f+p + iju/G+f+p + im/D+f+p + ima/D+f+p + e/Acc+f+p + e/Voc+f+p + im/L+f+p + ima/L+f+p + im/I+f+p + ima/I+f+p)

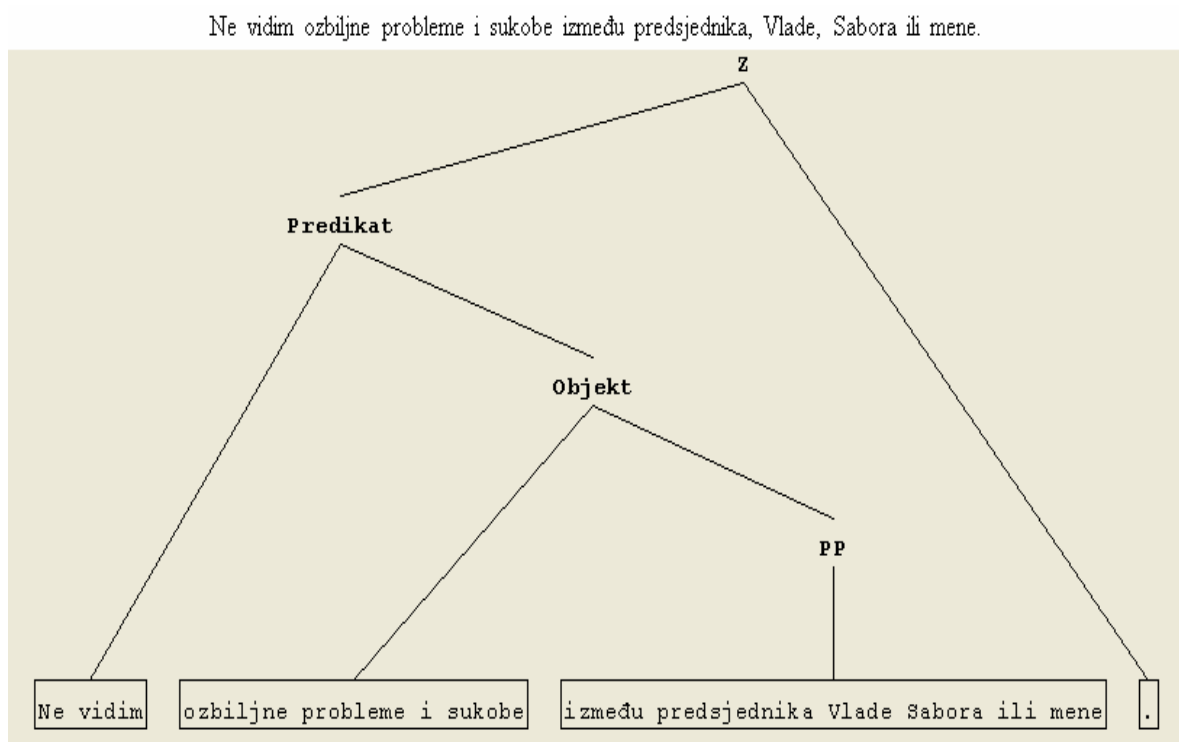
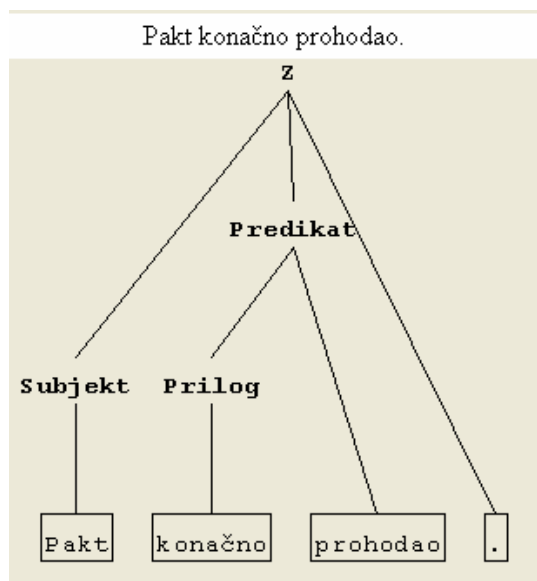
+<B2>v(e/Nom+n+s + ega/G+n+s + emu/D+n+s + e/Acc+n+s + e/Voc+n+s + em/L+n+s + emu/L+n+s + im/I+n+s + ime/I+n+s)

+<B2>v(a/Nom+n+p + ih/G+n+p + iju/G+n+p + im/D+n+p + ima/D+n+p + a/Acc+n+p + a/Voc+n+p + im/L+n+p + ima/L+n+p + im/I+n+p + ima/L+n+p);

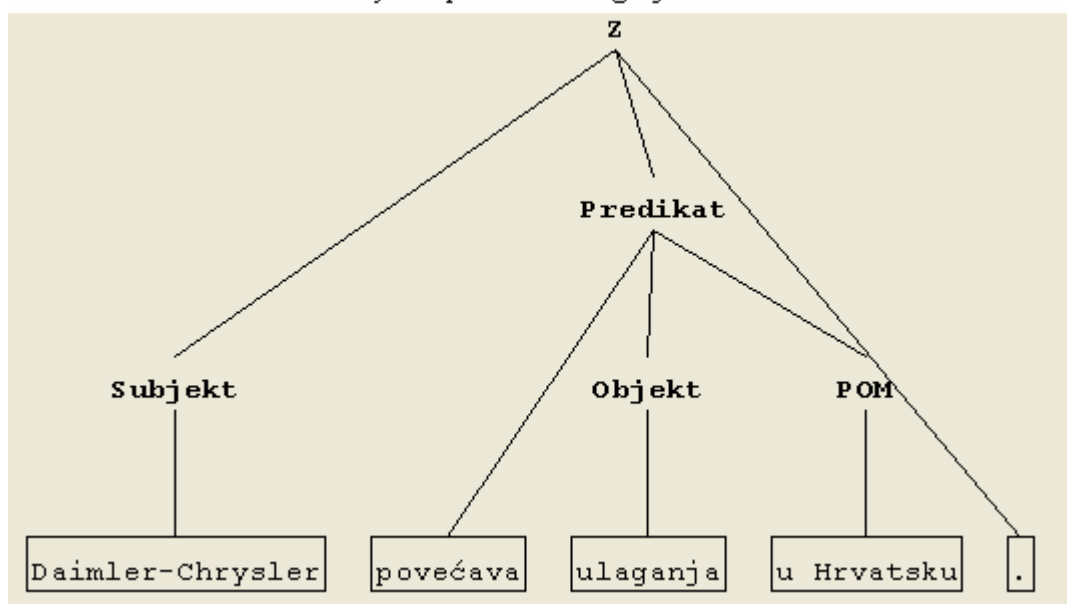


## Prilog 7 – Primjeri parsnih stabala

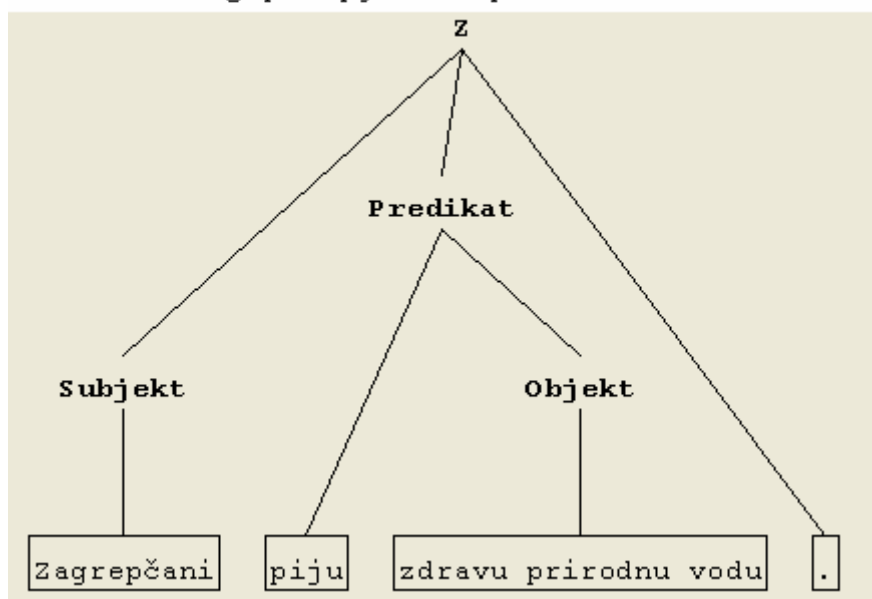
Primjeri parsnih stabala za prvih devet rečenica iz teksta uzetog za zlatni standard za testiranje parsera.



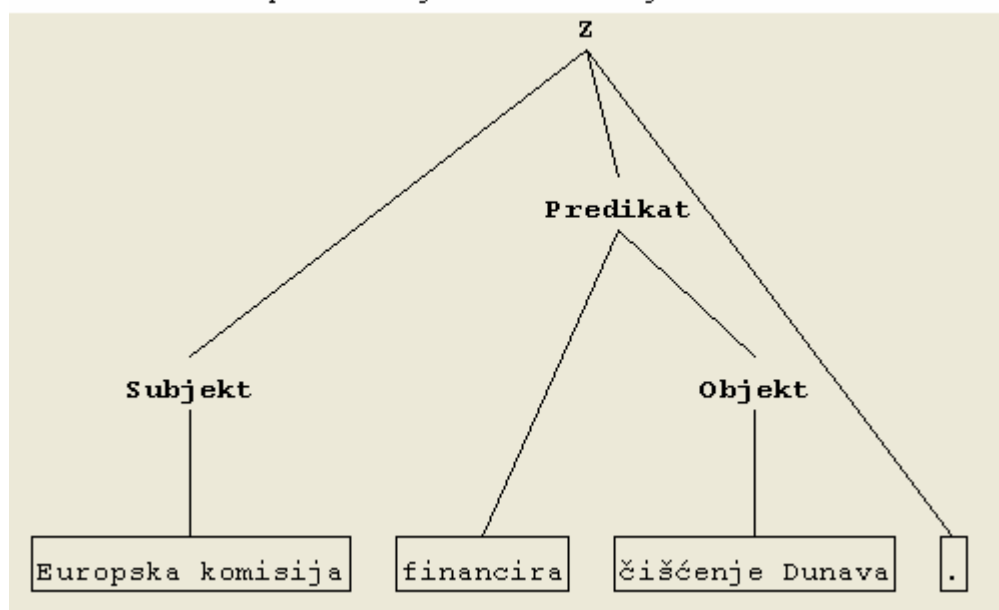
Daimler-Chrysler povećava ulaganja u Hrvatsku.



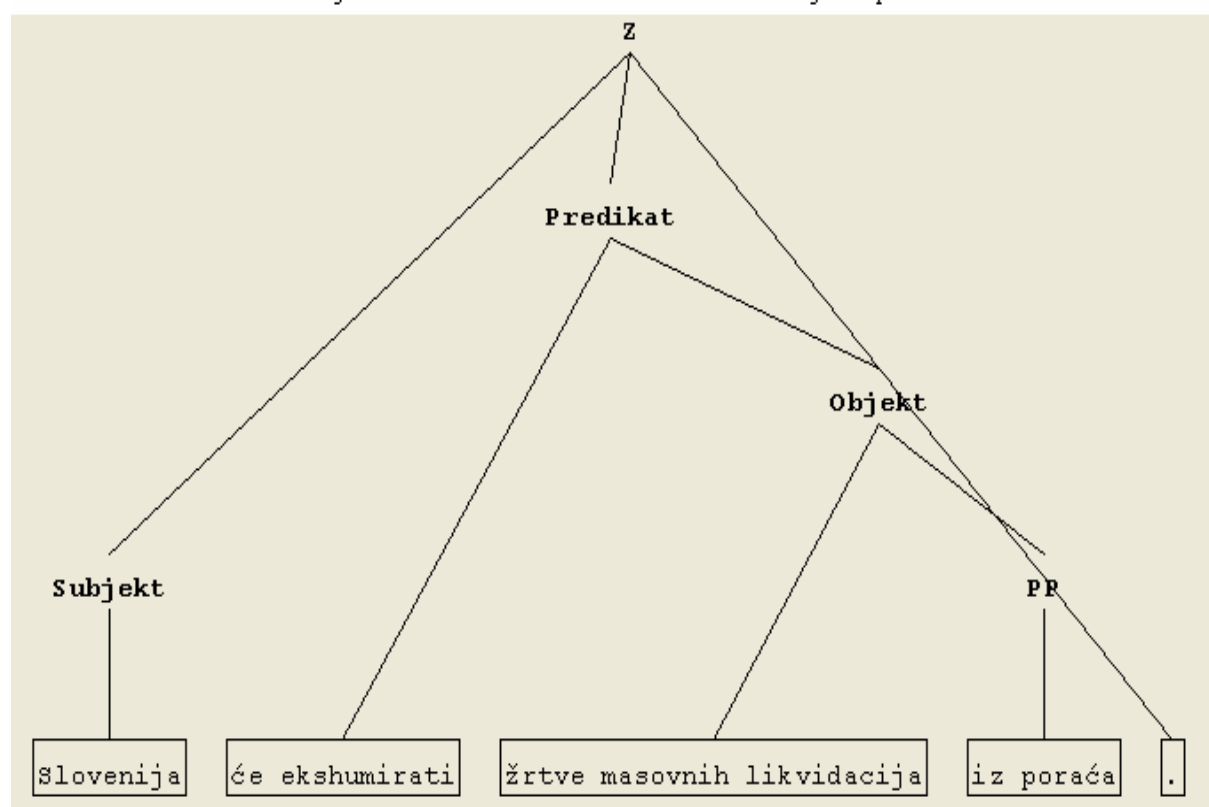
Zagrepčani piju zdravu prirodnu vodu.



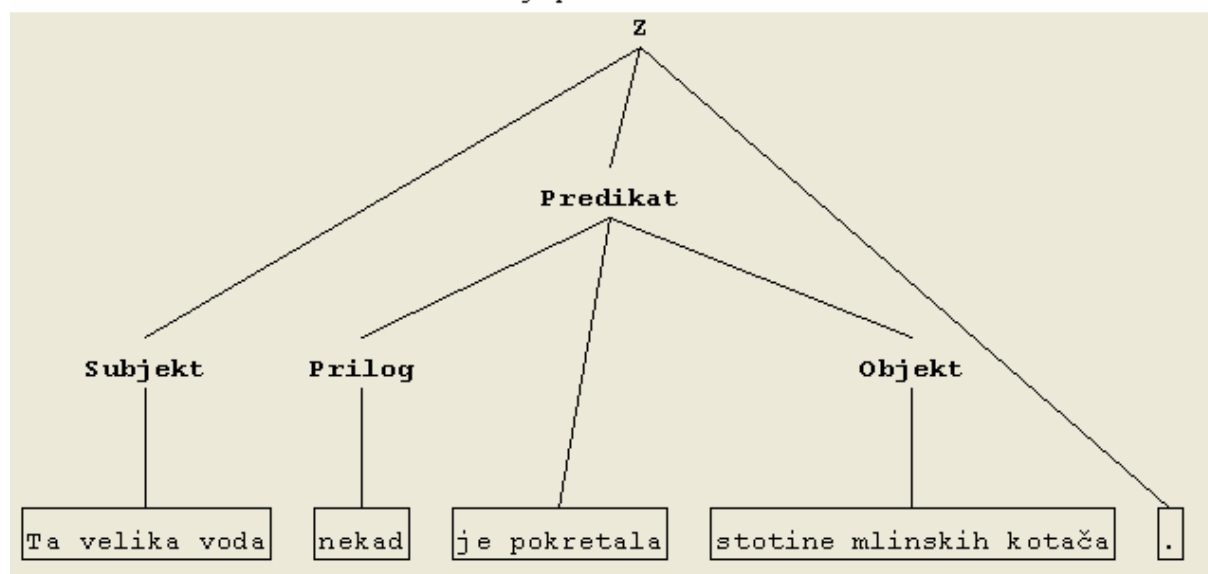
Europska komisija financira čišćenje Dunava.



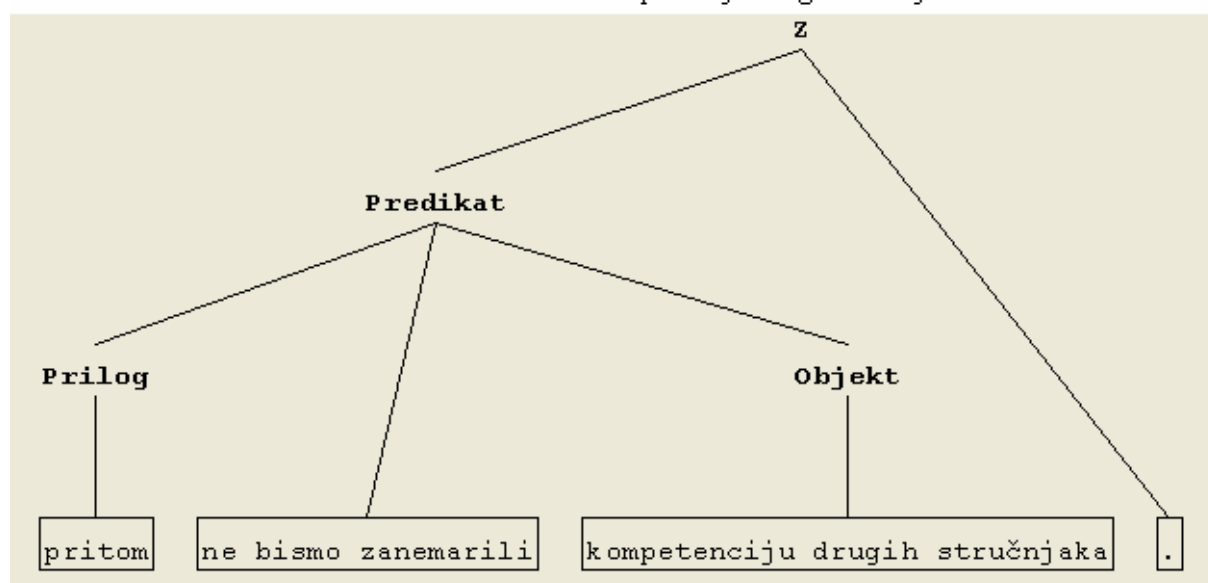
Slovenija će ekshumirati žrtve masovnih likvidacija iz poraća.

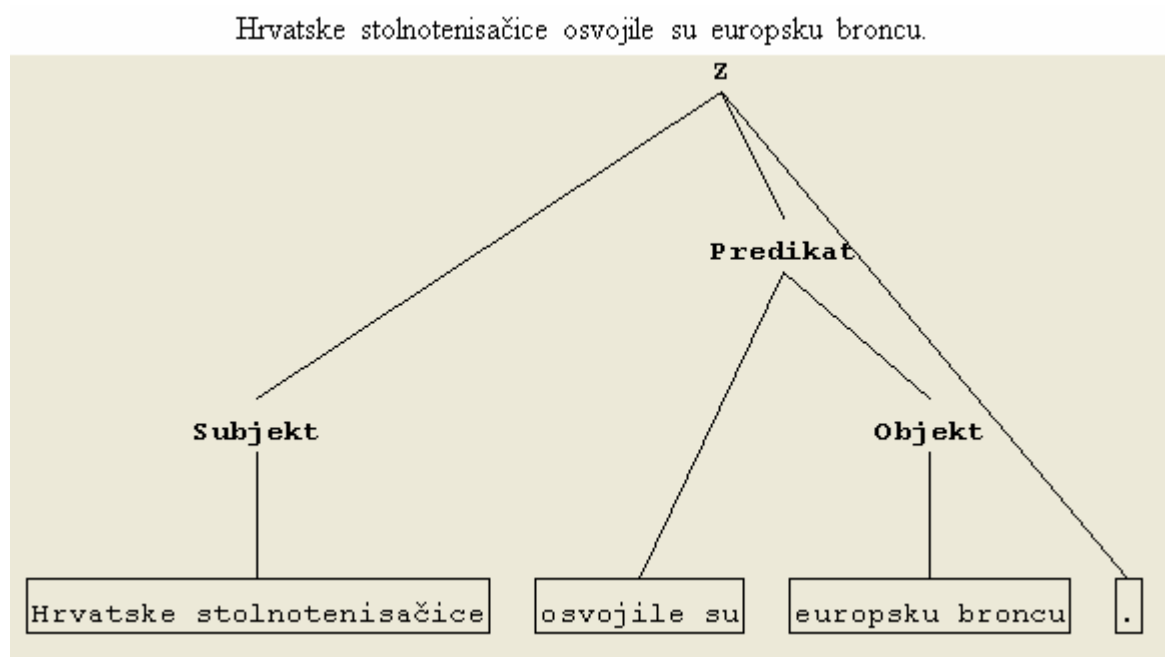


Ta velika voda nekad je pokretala stotine mlinskih kotača.



Pritom ne bismo zanemarili kompetenciju drugih stručnjaka.





## Prilog 8 – Popis slika

Slika 1: Grafički prikaz područja računalne lingvistike .....	8
Slika 2: Prikaz gramatike uz pomoć pravila .....	24
Slika 3: Prikaz uz pomoć stabla izvođenja .....	24
Slika 4: Negacija u izlazu .....	28
Slika 5: Negacija cijele klase .....	29
Slika 6: Negacija dijela klase .....	29
Slika 7: Prikaz u obliku mreže prijelaza .....	47
Slika 8: Prikaz u obliku stabla .....	29
Slika 9: Prikaz u obliku liste .....	29
Slika 10: Struktura parsera .....	51
Slika 11: Tip parsera prema sintaksoj strukturi .....	529
Slika 12: Grafički prikaz parsera upravljanog podacima .....	59
Slika 13: Navigacijski parser .....	75
Slika 14: Odnos vrsta riječi u zlatnom standardu za parser .....	92
Slika 15: Lingvistička analiza .....	94
Slika 16: Prikaz zapisa u rječniku .....	94
Slika 17: Ulazni i izlazni čvor .....	97
Slika 18: Sadržaj čvorova .....	97
Slika 19: Prikaz pretvarača .....	98
Slika 20: Sadržaj čvora je drugi graf .....	98
Slika 21: Prikaz varijable .....	98
Slika 22: Provjera vrijednosti atributa varijable .....	99
Slika 23: Radno sučelje u 'PrepareForNooJ' programu .....	99
Slika 24: Uspoređivanje nastavaka i dodavanje paradigme (FLX=x) .....	101
Slika 25: XML zapis CROVALLEX-a .....	102
Slika 26: Konverzija CROVALLEX XML zapisa u NooJ *.dic zapis .....	103
Slika 27: Glavni graf VP_NP_PP.nog .....	1024
Slika 28: Podgrafovi glavnog VP_NP_PP.nog grafa .....	1025
Slika 29: Osnovni <NP> graf .....	1026
Slika 30: Primjer za <NP> u nominativu .....	1027
Slika 31: Primjer za <NP> u akuzativu .....	1027
Slika 32: Glavni graf za prepoznavanje apozicije .....	120
Slika 33: Podgraf za prepoznavanje apozicije u akuzativu .....	10220
Slika 34: Podgraf za prepoznavanje vlastitih imena unutar grafa za apoziciju .....	122

Slika 35: Glavna gramatika za prepoznavanje atributa .....	122
Slika 36: Podgraf za prepoznavanje atributa u dativu .....	122
Slika 37: Glavni VP graf.....	125
Slika 38: Graf za prepoznavanje pluskvamperfekta povratnih i nepovratnih glagola .....	1224
Slika 39: Graf za prepoznavanje kondicionala I. povratnih i nepovratnih glagola .....	1226
Slika 40: Graf za prepoznavanje kondicionala II. povratnih i nepovratnih glagola .....	128
Slika 41: Graf za prepoznavanje imperativa povratnih i nepovratnih glagola .....	1228
Slika 42: Graf za prepoznavanje jednostavnih glagolskih oblika .....	131
Slika 43: Graf za prepoznavanje jednostavnih glagolskih oblika povratnih glagola .....	131
Slika 44: Graf za prepoznavanje perfekta .....	122
Slika 45: Graf za prepoznavanje pluskvamperfekta.....	12231
Slika 46: Glavni PP graf.....	12233
Slika 47: Graf za prepoznavanje <NP> dijela <PP> sintagme u lokativu .....	137
Slika 48: Dio grafa VP_NP_PP.nog .....	137
Slika 49: Brojevi.....	12236
Slika 50: Gramatika za opis rečenice .....	12237
Slika 51: <biti> + nominativna konstrukcija, <htjeti> + akuzativna konstrukcija .....	141
Slika 52: Gramatika VP_DCobl_PCobl .....	142
Slika 53: VP sintagma ima obaveznu NP dopunu u genitivu .....	143
Slika 54: VP sintagma ima obaveznu NP dopunu u dativu .....	144
Slika 55: VP sintagma s obaveznom dopunom u akuzativu i izbornom dopunom u nekom padežu .....	122
Slika 56: <NP>, <PP> ili <C> kao pojedinačne obavezne dopune glagolu.....	12243
Slika 57: Konkordancija za pojedinačne obavezne dopune glagolu .....	12244
Slika 58: <NP> ili <PP> kao pojedinačne neobavezne dopune glagolu .....	12244
Slika 59: Konkordancija za pojedinačne neobavezne dopune glagolu .....	12245
Slika 60: Subjekt_predikat.nog .....	12245
Slika 61: Gramatika za 'je' .....	150
Slika 62: Glavni graf za opis nepromjenjivih vrsta riječi .....	152
Slika 63: Podgraf za opis veznika .....	154
Slika 64: Podgraf za opis čestica <Q> .....	155
Slika 65: Podgraf za opis priloga <R> .....	12253
Slika 66: ObezlicenjeGlagola.nog .....	156

## SAŽETAK

U radu je opisan model djelomičnoga plitkog parsera upravljanog gramatikom čija je analiza prikazana linearno u obliku popisa singatmi koristeći se xml zapisom. Uz parser je opisan i prvi hrvatski razdjelnik. Pri izradi oba modela korištena je gramatika ovisnosti te kaskadno povezane lokalne gramatike kojima su se opisale imenske sintagme, apozicijske sintagme, atributske sintagme, glagolske sintagme i prijedložne sintagme.

Osnovne gramatike za opis navedenih sintagmi dodatno su specificirane korištenjem valencijskih okvira preuzetih iz hrvatskog valencijskog leksikona CROVALLEX. Pri tome je uspješno uklonjen veliki dio višeznačnosti imenskih sintagmi.

**Ključne riječi:** parsanje, djelomični parser, prirodni jezik, razdjelnik, gramatike, lokalne gramatike.



## SUMMARY

The paper describes the model of a partial shallow data driven parser. The analysis data is presented in a linear fashion using XML notation. Also, the first chunker for Croatian language is presented. Both models use dependency grammar and cascaded local grammars that describe noun chunks, apposition chunks, attribute chunks, verb chunks and prepositional chunks.

The main grammars for the recognition of chunks are further more enhanced with the description of verb valency frames that are described in the Croatian valency lexicon CROVALLEX. Large percentage of ambiguous noun chunks has been disambiguated in this way.

**Key words:** parsing, partial parser, natural language, chunker, grammar, local grammars.

## ŽIVOTOPIS

*Rođena sam 16.12.1973. u Metkoviću, R Hrvatska.*

*Prva tri razreda srednje škole završila sam u CUO "Petar Levantin" u Metkoviću. Četvrti razred srednje škole završila sam u "Sandalwood Senior Highschool" u Jacksonvillu, Florida, USA kao dvanaesta u klasi, čime sam ušla među 5% najboljih američkih srednjoškolaca.*

*1992. godine upisala sam se na FCCJ (Florida Community College at Jacksonville) i prvu godinu završila u klasi A studenata.*

*1993. godine prebacila sam se u Zagreb na Filozofski fakultet, Odsjek za informacijske znanosti i Engleski jezik i književnost. U studenom 1998. diplomirala sam i dobila stručni naziv profesora opće informatologije i engleskog jezika i književnosti.*

*Magistrirala sam u svibnju 2004. godine na poslijediplomskom studiju informacijskih znanosti s temom 'Padežne gramatike i razumijevanje hrvatskoga jezika'.*

*Od studenog 2000. godine radim kao znanstveni novak kod prof. dr. sc. Zdravka Dovedana, red. prof. na Odsjeku za informacijske znanosti.*